

QUESTIONS AND ANSWERS

1. What is database?

A database is a logically coherent collection of data with some inherent meaning, representing some aspect of real world and which is designed, built and populated with data for a specific purpose.

2. What is DBMS?

It is a collection of programs that enables user to create and maintain a database. In other words it is general-purpose software that provides the users with the processes of defining, constructing and manipulating the database for various applications.

3. What is a Database system?

The database and **DBMS** software together is called as Database system.

4. Advantages of DBMS?

- Ø Redundancy is controlled.
- Ø Unauthorised access is restricted.
- Ø Providing multiple user interfaces.
- Ø Enforcing integrity constraints.
- Ø Providing backup and recovery.

5. Disadvantage in File Processing System?

- Ø Data redundancy & inconsistency.
- Ø Difficult in accessing data.
- Ø Data isolation.
- Ø Data integrity.
- Ø Concurrent access is not possible.
- Ø Security Problems.

6. Describe the three levels of data abstraction?

The are three levels of abstraction:

- Ø Physical level: The lowest level of abstraction describes how data are stored.
- Ø Logical level: The next higher level of abstraction, describes what data are

stored in database and what relationship among those data.

∅ View level: The highest level of abstraction describes only part of entire database.

7. Define the "integrity rules"

There are two Integrity rules.

∅ Entity Integrity: States that "Primary key cannot have NULL value"

∅ Referential Integrity: States that "Foreign Key can be either a NULL value or should be Primary Key value of other relation.

8. What is extension and intension?

Extension -

It is the number of tuples present in a table at any instance. This is time dependent.

Intension -

It is a constant value that gives the name, structure of table and the constraints laid on it.

9. What is System R? What are its two major subsystems?

System R was designed and developed over a period of 1974-79 at IBM San Jose Research Center. It is a prototype and its purpose was to demonstrate that it is possible to build a Relational System that can be used in a real life environment to solve real life problems, with performance at least comparable to that of existing system.

Its two subsystems are

∅ Research Storage

∅ System Relational Data System.

10. How is the data structure of System R different from the relational structure?

Unlike Relational systems in System R

∅ Domains are not supported

∅ Enforcement of candidate key uniqueness is optional

∅ Enforcement of entity integrity is optional

∅ Referential integrity is not enforced

11. What is Data Independence?

Data independence means that "the application is independent of the storage structure and access strategy of data". In other words, The ability to modify the schema definition in one level should not affect the schema definition in

the next higher level.

Two types of Data Independence:

Ø Physical Data Independence: Modification in physical level should not affect the logical level.

Ø Logical Data Independence: Modification in logical level should affect the view level.

NOTE: Logical Data Independence is more difficult to achieve

12. What is a view? How it is related to data independence?

A view may be thought of as a virtual table, that is, a table that does not really exist in its own right but is instead derived from one or more underlying base table. In other words, there is no stored file that directly represents the view instead a definition of view is stored in data dictionary. Growth and restructuring of base tables is not reflected in views. Thus the view can insulate users from the effects of restructuring and growth in the database. Hence accounts for logical data independence.

13. What is Data Model?

A collection of conceptual tools for describing data, data relationships data semantics and constraints.

14. What is E-R model?

This data model is based on real world that consists of basic objects called entities and of relationship among these objects. Entities are described in a database by a set of attributes.

15. What is Object Oriented model?

This model is based on collection of objects. An object contains values stored in instance variables within the object. An object also contains bodies of code that operate on the object. These bodies of code are called methods. Objects that contain same types of values and the same methods are grouped together into classes.

16. What is an Entity?

It is a 'thing' in the real world with an independent existence.

17. What is an Entity type?

It is a collection (set) of entities that have same attributes.

18. What is an Entity set?

It is a collection of all entities of particular entity type in the database.

19. What is an Extension of entity type?

The collections of entities of a particular entity type are grouped together into an entity set.

20. What is Weak Entity set?

An entity set may not have sufficient attributes to form a primary key, and its primary key compromises of its partial key and primary key of its parent entity, then it is said to be Weak Entity set.

21. What is an attribute?

It is a particular property, which describes the entity.

22. What is a Relation Schema and a Relation?

A relation Schema denoted by $R(A_1, A_2, \dots, A_n)$ is made up of the relation name R and the list of attributes A_i that it contains. A relation is defined as a set of tuples. Let r be the relation which contains set tuples $(t_1, t_2, t_3, \dots, t_n)$. Each tuple is an ordered list of n -values $t=(v_1, v_2, \dots, v_n)$.

23. What is degree of a Relation?

It is the number of attribute of its relation schema.

24. What is Relationship?

It is an association among two or more entities.

25. What is Relationship set?

The collection (or set) of similar relationships.

26. What is Relationship type?

Relationship type defines a set of associations or a relationship set among a given set of entity types.

27. What is degree of Relationship type?

It is the number of entity type participating.

28. What is DDL (Data Definition Language)?

A data base schema is specifies by a set of definitions expressed by a special language called DDL.

29. What is VDL (View Definition Language)?

It specifies user views and their mappings to the conceptual schema.

30. What is SDL (Storage Definition Language)?

This language is to specify the internal schema. This language may specify the mapping between two schemas.

31. What is Data Storage - Definition Language?

The storage structures and access methods used by database system are specified by a set of definition in a special type of DDL called data storage-definition language.

32. What is DML (Data Manipulation Language)?

This language that enable user to access or manipulate data as organised by appropriate data model.

Ø Procedural DML or Low level: DML requires a user to specify what data are needed and how to get those data.

Ø Non-Procedural DML or High level: DML requires a user to specify what data are needed without specifying how to get those data.

33. What is DML Compiler?

It translates DML statements in a query language into low-level instruction that the query evaluation engine can understand.

34. What is Query evaluation engine?

It executes low-level instruction generated by compiler.

35. What is DDL Interpreter?

It interprets DDL statements and record them in tables containing metadata.

36. What is Record-at-a-time?

The Low level or Procedural DML can specify and retrieve each record from a set of records. This retrieve of a record is said to be Record-at-a-time.

37. What is Set-at-a-time or Set-oriented?

The High level or Non-procedural DML can specify and retrieve many records in a single DML statement. This retrieve of a record is said to be Set-at-a-time or Set-oriented.

38. What is Relational Algebra?

It is procedural query language. It consists of a set of operations that take one or two relations as input and produce a new relation.

39. What is Relational Calculus?

It is an applied predicate calculus specifically tailored for relational databases proposed by E.F. Codd. E.g. of languages based on it are DSL ALPHA, QUEL.

40. How does Tuple-oriented relational calculus differ from domain-oriented relational calculus

The tuple-oriented calculus uses a tuple variables i.e., variable whose only permitted values are tuples of that relation. E.g. QUEL

The domain-oriented calculus has domain variables i.e., variables that range over the underlying domains instead of over relation. E.g. ILL, DEDUCE.

41. What is normalization?

It is a process of analysing the given relation schemas based on their Functional Dependencies (FDs) and primary key to achieve the properties

Ø Minimizing redundancy

Ø Minimizing insertion, deletion and update anomalies.

42. What is Functional Dependency?

A Functional dependency is denoted by $X \rightarrow Y$ between two sets of attributes X and Y that are subsets of R specifies a constraint on the possible tuple that can form a relation state r of R . The constraint is for any two tuples t_1 and t_2 in r if $t_1[X] = t_2[X]$ then they have $t_1[Y] = t_2[Y]$. This means the value of X component of a tuple uniquely determines the value of component Y .

43. When is a functional dependency F said to be minimal?

Ø Every dependency in F has a single attribute for its right hand side.

Ø We cannot replace any dependency $X \rightarrow A$ in F with a dependency $Y \rightarrow A$ where Y is a proper subset of X and still have a set of dependency that is equivalent to F .

Ø We cannot remove any dependency from F and still have set of dependency that is equivalent to F .

44. What is Multivalued dependency?

Multivalued dependency denoted by $X \twoheadrightarrow Y$ specified on relation schema R , where X and Y are both subsets of R , specifies the following constraint on

any relation r of R : if two tuples t_1 and t_2 exist in r such that $t_1[X] = t_2[X]$ then t_3 and t_4 should also exist in r with the following properties

$\emptyset t_3[X] = t_4[X] = t_1[X] = t_2[X]$

$\emptyset t_3[Y] = t_1[Y]$ and $t_4[Y] = t_2[Y]$

$\emptyset t_3[Z] = t_2[Z]$ and $t_4[Z] = t_1[Z]$

where $[Z = (R - (X \cup Y))]$

45. What is Lossless join property?

It guarantees that the spurious tuple generation does not occur with respect to relation schemas after decomposition.

46. What is 1 NF (Normal Form)?

The domain of attribute must include only atomic (simple, indivisible) values.

47. What is Fully Functional dependency?

It is based on concept of full functional dependency. A functional dependency $X \twoheadrightarrow Y$ is full functional dependency if removal of any attribute A from X means that the dependency does not hold any more.

48. What is 2NF?

A relation schema R is in 2NF if it is in 1NF and every non-prime attribute A in R is fully functionally dependent on primary key.

49. What is 3NF?

A relation schema R is in 3NF if it is in 2NF and for every FD $X \twoheadrightarrow A$ either of the following is true

$\emptyset X$ is a Super-key of R .

$\emptyset A$ is a prime attribute of R .

In other words, if every non prime attribute is non-transitively dependent on primary key.

50. What is BCNF (Boyce-Codd Normal Form)?

A relation schema R is in BCNF if it is in 3NF and satisfies an additional constraint that for every FD $X \twoheadrightarrow A$, X must be a candidate key.

51. What is 4NF?

A relation schema R is said to be in 4NF if for every Multivalued dependency $X \twoheadrightarrow Y$ that holds over R , one of following is true

$\emptyset X$ is subset or equal to (or) $XY = R$.

Ø X is a super key.

52. What is 5NF?

A Relation schema R is said to be 5NF if for every join dependency {R1, R2, ..., Rn} that holds R, one the following is true

Ø Ri = R for some i.

Ø The join dependency is implied by the set of FD, over R in which the left side is key of R.

53. What is Domain-Key Normal Form?

A relation is said to be in DKNF if all constraints and dependencies that should hold on the the constraint can be enforced by simply enforcing the domain constraint and key constraint on the relation.

54. What are partial, alternate,, artificial, compound and natural key?

Partial Key:

It is a set of attributes that can uniquely identify weak entities and that are related to same owner entity. It is sometime called as Discriminator.

Alternate Key:

All Candidate Keys excluding the Primary Key are known as Alternate Keys.

Artificial Key:

If no obvious key, either stand alone or compound is available, then the last resort is to simply create a key, by assigning a unique number to each record or occurrence. Then this is known as developing an artificial key.

Compound Key:

If no single data element uniquely identifies occurrences within a construct, then combining multiple elements to create a unique identifier for the construct is known as creating a compound key.

Natural Key:

When one of the data elements stored within a construct is utilized as the primary key, then it is called the natural key.

55. What is indexing and what are the different kinds of indexing?

Indexing is a technique for determining how quickly specific data can be found.

Types:

Ø Binary search style indexing

Ø B-Tree indexing

Ø Inverted list indexing

- Ø Memory resident table
- Ø Table indexing

56. What is system catalog or catalog relation? How is better known as?

A RDBMS maintains a description of all the data that it contains, information about every relation and index that it contains. This information is stored in a collection of relations maintained by the system called metadata. It is also called data dictionary.

57. What is meant by query optimization?

The phase that identifies an efficient execution plan for evaluating a query that has the least estimated cost is referred to as query optimization.

58. What is join dependency and inclusion dependency?

Join Dependency:

A Join dependency is generalization of Multivalued dependency. A JD $\{R_1, R_2, \dots, R_n\}$ is said to hold over a relation R if $R_1, R_2, R_3, \dots, R_n$ is a lossless-join decomposition of R. There is no set of sound and complete inference rules for JD.

Inclusion Dependency:

An Inclusion Dependency is a statement of the form that some columns of a relation are contained in other columns. A foreign key constraint is an example of inclusion dependency.

59. What is durability in DBMS?

Once the **DBMS** informs the user that a transaction has successfully completed, its effects should persist even if the system crashes before all its changes are reflected on disk. This property is called durability.

60. What do you mean by atomicity and aggregation?

Atomicity:

Either all actions are carried out or none are. Users should not have to worry about the effect of incomplete transactions. **DBMS** ensures this by undoing the actions of incomplete transactions.

Aggregation:

A concept which is used to model a relationship between a collection of entities and relationships. It is used when we need to express a relationship among relationships.

61. What is a Phantom Deadlock?

In distributed deadlock detection, the delay in propagating local information

might cause the deadlock detection algorithms to identify deadlocks that do not really exist. Such situations are called phantom deadlocks and they lead to unnecessary aborts.

62. What is a checkpoint and When does it occur?

A Checkpoint is like a snapshot of the **DBMS** state. By taking checkpoints, the **DBMS** can reduce the amount of work to be done during restart in the event of subsequent crashes.

63. What are the different phases of transaction?

Different phases are

- Ø Analysis phase
- Ø Redo Phase
- Ø Undo phase

64. What do you mean by flat file database?

It is a database in which there are no programs or user access languages. It has no cross-file capabilities but is user-friendly and provides user-interface management.

65. What is "transparent DBMS"?

It is one, which keeps its Physical Structure hidden from user.

66. Brief theory of Network, Hierarchical schemas and their properties

Network schema uses a graph data structure to organize records example for such a database management system is CTCG while a hierarchical schema uses a tree data structure example for such a system is IMS.

67. What is a query?

A query with respect to **DBMS** relates to user commands that are used to interact with a data base. The query language can be classified into data definition language and data manipulation language.

68. What do you mean by Correlated subquery?

Subqueries, or nested queries, are used to bring back a set of rows to be used by the parent query. Depending on how the subquery is written, it can be executed once for the parent query or it can be executed once for each row returned by the parent query. If the subquery is executed for each row of the parent, this is called a correlated subquery.

A correlated subquery can be easily identified if it contains any references to

the parent subquery columns in its WHERE clause. Columns from the subquery cannot be referenced anywhere else in the parent query. The following example demonstrates a non-correlated subquery.

E.g. Select * From CUST Where '10/03/1990' IN (Select ODATE From ORDER Where CUST.CNUM = ORDER.CNUM)

69. What are the primitive operations common to all record management systems?

Addition, deletion and modification.

70. Name the buffer in which all the commands that are typed in are stored

'Edit' Buffer

71. What are the unary operations in Relational Algebra?

PROJECTION and SELECTION.

72. Are the resulting relations of PRODUCT and JOIN operation the same?

No.

PRODUCT: Concatenation of every row in one relation with every row in another.

JOIN: Concatenation of rows from one relation and related rows from another.

73. What is RDBMS KERNEL?

Two important pieces of RDBMS architecture are the kernel, which is the software, and the data dictionary, which consists of the system-level data structures used by the kernel to manage the database

You might think of an RDBMS as an operating system (or set of subsystems), designed specifically for controlling data access; its primary functions are storing, retrieving, and securing data. An RDBMS maintains its own list of authorized users and their associated privileges; manages memory caches and paging; controls locking for concurrent resource usage; dispatches and schedules user requests; and manages space usage within its table-space structures.

74. Name the sub-systems of a RDBMS

I/O, Security, Language Processing, Process Control, Storage Management, Logging and Recovery, Distribution Control, Transaction Control, Memory

Management, Lock Management

75. Which part of the RDBMS takes care of the data dictionary? How

Data dictionary is a set of tables and database objects that is stored in a special area of the database and maintained exclusively by the kernel.

76. What is the job of the information stored in data-dictionary?

The information in the data dictionary validates the existence of the objects, provides access to them, and maps the actual physical storage location.

77. Not only RDBMS takes care of locating data it also

determines an optimal access path to store or retrieve the data

76. How do you communicate with an RDBMS?

You communicate with an RDBMS using Structured Query Language (SQL)

78. Define SQL and state the differences between SQL and other conventional programming Languages

SQL is a nonprocedural language that is designed specifically for data access operations on normalized relational database structures. The primary difference between SQL and other conventional programming languages is that SQL statements specify what data operations should be performed rather than how to perform them.

79. Name the three major set of files on disk that compose a database in Oracle

There are three major sets of files on disk that compose a database. All the files are binary. These are

Ø Database files

Ø Control files

Ø Redo logs

The most important of these are the database files where the actual data resides. The control files and the redo logs support the functioning of the architecture itself.

All three sets of files must be present, open, and available to Oracle for any data on the database to be useable. Without these files, you cannot access the database, and the database administrator might have to recover some or all of the database using a backup, if there is one.

80. What is an Oracle Instance?

The Oracle system processes, also known as Oracle background processes, provide functions for the user processes—functions that would otherwise be done by the user processes themselves

Oracle database-wide system memory is known as the SGA, the system global area or shared global area. The data and control structures in the SGA are shareable, and all the Oracle background processes and user processes can use them.

The combination of the SGA and the Oracle background processes is known as an Oracle instance

81. What are the four Oracle system processes that must always be up and running for the database to be useable

The four Oracle system processes that must always be up and running for the database to be useable include DBWR (Database Writer), LGWR (Log Writer), SMON (System Monitor), and PMON (Process Monitor).

82. What are database files, control files and log files. How many of these files should a database have at least? Why?

Database Files

The database files hold the actual data and are typically the largest in size. Depending on their sizes, the tables (and other objects) for all the user accounts can go in one database file—but that's not an ideal situation because it does not make the database structure very flexible for controlling access to storage for different users, putting the database on different disk drives, or backing up and restoring just part of the database.

You must have at least one database file but usually, more than one files are used. In terms of accessing and using the data in the tables and other objects, the number (or location) of the files is immaterial.

The database files are fixed in size and never grow bigger than the size at which they were created

Control Files

The control files and redo logs support the rest of the architecture. Any database must have at least one control file, although you typically have more than one to guard against loss. The control file records the name of the database, the date and time it was created, the location of the database and redo logs, and the synchronization information to ensure that all three sets of files are always in step. Every time you add a new database or redo log file to the database, the information is recorded in the control files.

Redo Logs

Any database must have at least two redo logs. These are the journals for the

database; the redo logs record all changes to the user objects or system objects. If any type of failure occurs, the changes recorded in the redo logs can be used to bring the database to a consistent state without losing any committed transactions. In the case of non-data loss failure, Oracle can apply the information in the redo logs automatically without intervention from the DBA.

The redo log files are fixed in size and never grow dynamically from the size at which they were created.

83. What is ROWID?

The ROWID is a unique database-wide physical address for every row on every table. Once assigned (when the row is first inserted into the database), it never changes until the row is deleted or the table is dropped.

The ROWID consists of the following three components, the combination of which uniquely identifies the physical storage location of the row.

Ø Oracle database file number, which contains the block with the rows

Ø Oracle block address, which contains the row

Ø The row within the block (because each block can hold many rows)

The ROWID is used internally in indexes as a quick means of retrieving rows with a particular key value. Application developers also use it in SQL statements as a quick way to access a row once they know the ROWID

84. What is Oracle Block? Can two Oracle Blocks have the same address?

Oracle "formats" the database files into a number of Oracle blocks when they are first created—making it easier for the RDBMS software to manage the files and easier to read data into the memory areas.

The block size should be a multiple of the operating system block size.

Regardless of the block size, the entire block is not available for holding data; Oracle takes up some space to manage the contents of the block. This block header has a minimum size, but it can grow.

These Oracle blocks are the smallest unit of storage. Increasing the Oracle block size can improve performance, but it should be done only when the database is first created.

Each Oracle block is numbered sequentially for each database file starting at 1. Two blocks can have the same block address if they are in different database files.

85. What is database Trigger?

A database trigger is a PL/SQL block that can be defined to automatically execute for insert, update, and delete statements against a table. The trigger can be defined to execute once for the entire statement or once for every row that is inserted, updated, or deleted. For any one table, there are twelve events for which you can define database triggers. A database trigger can call database procedures that are also written in PL/SQL.

86. Name two utilities that Oracle provides, which are used for backup and recovery.

Along with the RDBMS software, Oracle provides two utilities that you can use to back up and restore the database. These utilities are Export and Import.

The Export utility dumps the definitions and data for the specified part of the database to an operating system binary file. The Import utility reads the file produced by an export, recreates the definitions of objects, and inserts the data.

If Export and Import are used as a means of backing up and recovering the database, all the changes made to the database cannot be recovered since the export was performed. The best you can do is recover the database to the time when the export was last performed.

87. What are stored-procedures? And what are the advantages of using them.

Stored procedures are database objects that perform a user defined operation. A stored procedure can have a set of compound SQL statements. A stored procedure executes the SQL commands and returns the result to the client. Stored procedures are used to reduce network traffic.

88. How are exceptions handled in PL/SQL? Give some of the internal exceptions' name

PL/SQL exception handling is a mechanism for dealing with run-time errors encountered during procedure execution. Use of this mechanism enables execution to continue if the error is not severe enough to cause procedure termination.

The exception handler must be defined within a subprogram specification. Errors cause the program to raise an exception with a transfer of control to the exception-handler block. After the exception handler executes, control

returns to the block in which the handler was defined. If there are no more executable statements in the block, control returns to the caller.

User-Defined Exceptions

PL/SQL enables the user to define exception handlers in the declarations area of subprogram specifications. User accomplishes this by naming an exception as in the following example:

```
ot_failure EXCEPTION;
```

In this case, the exception name is `ot_failure`. Code associated with this handler is written in the `EXCEPTION` specification area as follows:

```
EXCEPTION
when OT_FAILURE then
out_status_code := g_out_status_code;
out_msg := g_out_msg;
```

The following is an example of a subprogram exception:

```
EXCEPTION
when NO_DATA_FOUND then
g_out_status_code := 'FAIL';
RAISE ot_failure;
```

Within this exception is the `RAISE` statement that transfers control back to the `ot_failure` exception handler. This technique of raising the exception is used to invoke all user-defined exceptions.

System-Defined Exceptions

Exceptions internal to PL/SQL are raised automatically upon error.

`NO_DATA_FOUND` is a system-defined exception. Table below gives a complete list of internal exceptions.

PL/SQL internal exceptions.

PL/SQL internal exceptions.

Exception Name	Oracle Error
<code>CURSOR_ALREADY_OPEN</code>	ORA-06511
<code>DUP_VAL_ON_INDEX</code>	ORA-00001
<code>INVALID_CURSOR</code>	ORA-01001
<code>INVALID_NUMBER</code>	ORA-01722
<code>LOGIN_DENIED</code>	ORA-01017
<code>NO_DATA_FOUND</code>	ORA-01403
<code>NOT_LOGGED_ON</code>	ORA-01012
<code>PROGRAM_ERROR</code>	ORA-06501
<code>STORAGE_ERROR</code>	ORA-06500

TIMEOUT_ON_RESOURCE ORA-00051
TOO_MANY_ROWS ORA-01422
TRANSACTION_BACKED_OUT ORA-00061
VALUE_ERROR ORA-06502
ZERO_DIVIDE ORA-01476

In addition to this list of exceptions, there is a catch-all exception named OTHERS that traps all errors for which specific error handling has not been established.

89. Does PL/SQL support "overloading"? Explain

The concept of overloading in PL/SQL relates to the idea that you can define procedures and functions with the same name. PL/SQL does not look only at the referenced name, however, to resolve a procedure or function call. The count and data types of formal parameters are also considered.

PL/SQL also attempts to resolve any procedure or function calls in locally defined packages before looking at globally defined packages or internal functions. To further ensure calling the proper procedure, you can use the dot notation. Prefacing a procedure or function name with the package name fully qualifies any procedure or function reference.

90. Tables derived from the ERD

- a) Are totally unnormalised
- b) Are always in 1NF
- c) Can be further denormalised
- d) May have multi-valued attributes

(b) Are always in 1NF

91. Spurious tuples may occur due to

- i. Bad normalization
 - ii. Theta joins
 - iii. Updating tables from join
- a) i & ii b) ii & iii
c) i & iii d) ii & iii

(a) i & iii because theta joins are joins made on keys that are not primary keys.

92. A B C is a set of attributes. The functional dependency is as follows

AB -> B

AC -> C

C -> B

- a) is in 1NF
- b) is in 2NF
- c) is in 3NF
- d) is in BCNF

(a) is in 1NF since $(AC)^+ = \{ A, B, C \}$ hence AC is the primary key. Since C B is a FD given, where neither C is a Key nor B is a prime attribute, this it is not in 3NF. Further B is not functionally dependent on key AC thus it is not in 2NF. Thus the given FDs is in 1NF.

93. In mapping of ERD to DFD

- a) entities in ERD should correspond to an existing entity/store in DFD
- b) entity in DFD is converted to attributes of an entity in ERD
- c) relations in ERD has 1 to 1 correspondence to processes in DFD
- d) relationships in ERD has 1 to 1 correspondence to flows in DFD

(a) entities in ERD should correspond to an existing entity/store in DFD

94. A dominant entity is the entity

- a) on the N side in a 1 : N relationship
- b) on the 1 side in a 1 : N relationship
- c) on either side in a 1 : 1 relationship
- d) nothing to do with 1 : 1 or 1 : N relationship

(b) on the 1 side in a 1 : N relationship

95. Select 'NORTH', CUSTOMER From CUST_DTLS Where

REGION = 'N' Order By

CUSTOMER Union Select 'EAST', CUSTOMER From CUST_DTLS

Where REGION = 'E' Order By CUSTOMER

The above is

- a) Not an error
- b) Error - the string in single quotes 'NORTH' and 'SOUTH'
- c) Error - the string should be in double quotes
- d) Error - ORDER BY clause

(d) Error - the ORDER BY clause. Since ORDER BY clause cannot be used in UNIONS

96. What is Storage Manager?

It is a program module that provides the interface between the low-level data stored in database, application programs and queries submitted to the system.

97. What is Buffer Manager?

It is a program module, which is responsible for fetching data from disk storage into main memory and deciding what data to be cache in memory.

98. What is Transaction Manager?

It is a program module, which ensures that database, remains in a consistent state despite system failures and concurrent transaction execution proceeds without conflicting.

99. What is File Manager?

It is a program module, which manages the allocation of space on disk storage and data structure used to represent information stored on a disk.

100. What is Authorization and Integrity manager?

It is the program module, which tests for the satisfaction of integrity constraint and checks the authority of user to access data.

101. What are stand-alone procedures?

Procedures that are not part of a package are known as stand-alone because they independently defined. A good example of a stand-alone procedure is one written in a SQL*Forms application. These types of procedures are not available for reference from other Oracle tools. Another limitation of stand-alone procedures is that they are compiled at run time, which slows execution.

102. What are cursors give different types of cursors.

PL/SQL uses cursors for all database information accesses statements. The language supports the use two types of cursors

Ø Implicit

Ø Explicit

103. What is cold backup and hot backup (in case of Oracle)?

Ø Cold Backup:

It is copying the three sets of files (database files, redo logs, and control file) when the instance is shut down. This is a straight file copy, usually from the disk directly to tape. You must shut down the instance to guarantee a consistent copy.

If a cold backup is performed, the only option available in the event of data file loss is restoring all the files from the latest backup. All work performed on the database since the last backup is lost.

Ø Hot Backup:

Some sites (such as worldwide airline reservations systems) cannot shut down the database while making a backup copy of the files. The cold backup is not an available option.

So different means of backing up database must be used — the hot backup. Issue a SQL command to indicate to Oracle, on a tablespace-by-tablespace basis, that the files of the tablespace are to be backed up. The users can continue to make full use of the files, including making changes to the data. Once the user has indicated that he/she wants to back up the tablespace files, he/she can use the operating system to copy those files to the desired backup destination.

The database must be running in ARCHIVELOG mode for the hot backup option.

If a data loss failure does occur, the lost database files can be restored using the hot backup and the online and offline redo logs created since the backup was done. The database is restored to the most consistent state without any loss of committed transactions.

104. What are Armstrong rules? How do we say that they are complete and/or sound

The well-known inference rules for FDs

Ø Reflexive rule :

If Y is subset or equal to X then X Y.

Ø Augmentation rule:

If X Y then XZ YZ.

Ø Transitive rule:

If {X Y, Y Z} then X Z.

Ø Decomposition rule :

If X YZ then X Y.

Ø Union or Additive rule:

If {X Y, X Z} then X YZ.

∅ Pseudo Transitive rule :

If $\{X \rightarrow Y, WY \rightarrow Z\}$ then $WX \rightarrow Z$.

Of these the first three are known as Armstrong Rules. They are sound because it is enough if a set of FDs satisfy these three. They are called complete because using these three rules we can generate the rest all inference rules.

105. How can you find the minimal key of relational schema?

Minimal key is one which can identify each tuple of the given relation schema uniquely. For finding the minimal key it is required to find the closure that is the set of all attributes that are dependent on any given set of attributes under the given set of functional dependency.

Algo. I Determining X^+ , closure for X, given set of FDs F

1. Set $X^+ = X$
2. Set Old $X^+ = X^+$
3. For each FD $Y \rightarrow Z$ in F and if Y belongs to X^+ then add Z to X^+
4. Repeat steps 2 and 3 until Old $X^+ = X^+$

Algo.II Determining minimal K for relation schema R, given set of FDs F

1. Set K to R that is make K a set of all attributes in R
2. For each attribute A in K
 - a. Compute $(K - A)^+$ with respect to F
 - b. If $(K - A)^+ = R$ then set $K = (K - A)^+$

106. What do you understand by dependency preservation?

Given a relation R and a set of FDs F, dependency preservation states that the closure of the union of the projection of F on each decomposed relation R_i is equal to the closure of F. i.e.,

$$((PR_1(F)) \cup \dots \cup (PR_n(F)))^+ = F^+$$

if decomposition is not dependency preserving, then some dependency is lost in the decomposition.

107. What is meant by Proactive, Retroactive and Simultaneous Update.

Proactive Update:

The updates that are applied to database before it becomes effective in real world .

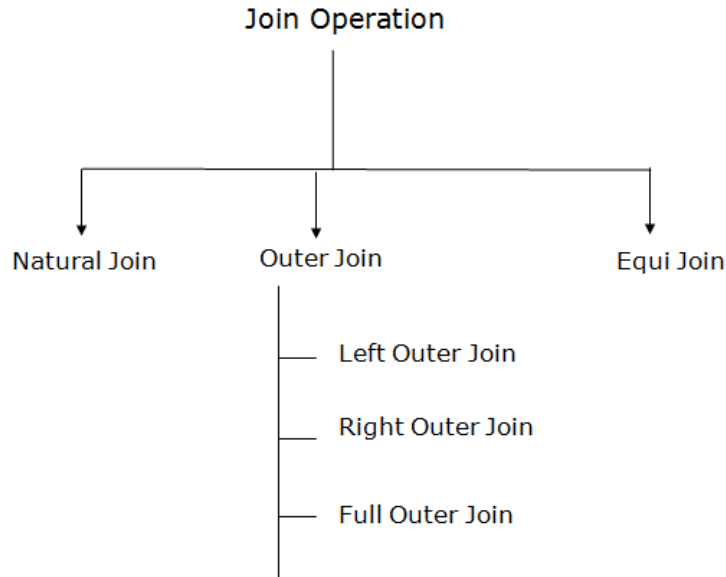
Retroactive Update:

The updates that are applied to database after it becomes effective in real world .

Simultaneous Update:

The updates that are applied to database at the same time when it becomes effective in real world .

108. What are the different types of JOIN operations?



109. What are some of the advantages of MongoDB?

Some advantages of MongoDB are as follows:

- MongoDB supports field, range-based, string pattern matching type queries. for searching the data in the database
- MongoDB support primary and secondary index on any fields
- MongoDB basically uses JavaScript objects in place of procedures
- MongoDB uses a dynamic database schema
- MongoDB is very easy to scale up or down
- MongoDB has inbuilt support for data partitioning (Sharding).

110. What is a Document in MongoDB?

A Document in MongoDB is an ordered set of keys with associated values. It is represented by a map, hash, or dictionary. In JavaScript, documents are represented as objects:

```
{"greeting" : "Hello world!"}
```

Complex documents will contain multiple key/value pairs:

```
{"greeting" : "Hello world!", "views" : 3}
```

111. What is a Collection in MongoDB?

A collection in MongoDB is a group of documents. If a document is the MongoDB analog of a row in a relational database, then a collection can be thought of as the analog to a table.

Documents within a single collection can have any number of different "shapes.", i.e. collections have dynamic schemas.

For example, both of the following documents could be stored in a single collection:

```
{"greeting" : "Hello world!", "views": 3}
{"signoff": "Good bye"}
```

112. What are Databases in MongoDB?

MongoDB groups collections into databases. MongoDB can host several databases, each grouping together collections.

Some reserved database names are as follows:

admin

local

config

113. What is the Mongo Shell?

It is a JavaScript shell that allows interaction with a MongoDB instance from the command line. With that one can perform administrative functions, inspecting an instance, or exploring MongoDB.

To start the shell, run the mongo executable:

```
$ mongod
$ mongo
MongoDB shell version: 4.2.0
connecting to: test
>
```

The shell is a full-featured JavaScript interpreter, capable of running arbitrary JavaScript programs. Let's see how basic math works on this:

```
> x = 100;
```

```
200
> x / 5;
20
```

114. How does Scale-Out occur in MongoDB?

The document-oriented data model of MongoDB makes it easier to split data across multiple servers. Balancing and loading data across a cluster is done by MongoDB. It then redistributes documents automatically.

The mongos acts as a query router, providing an interface between client applications and the sharded cluster.

Config servers store metadata and configuration settings for the cluster. MongoDB uses the config servers to manage distributed locks. Each sharded cluster must have its own config servers.

115. What are some features of MongoDB?

- **Indexing:** It supports generic secondary indexes and provides unique, compound, geospatial, and full-text indexing capabilities as well.
- **Aggregation:** It provides an aggregation framework based on the concept of data processing pipelines.
- **Special collection and index types:** It supports time-to-live (TTL) collections for data that should expire at a certain time
- **File storage:** It supports an easy-to-use protocol for storing large files and file metadata.
- **Sharding:** Sharding is the process of splitting data up across machines.

116. How to add data in MongoDB?

The basic method for adding data to MongoDB is "inserts". To insert a single document, use the collection's `insertOne` method:

```
> db.books.insertOne({"title" : "Start With Why"})
```

For inserting multiple documents into a collection, we use `insertMany`. This method enables passing an array of documents to the database.

117. How do you Update a Document?

Once a document is stored in the database, it can be changed using one of several update methods: `updateOne`, `updateMany`, and `replaceOne`. `updateOne` and `updateMany` each takes a filter document as their first parameter and a modifier document, which describes changes to make, as the second parameter. `replaceOne` also takes a filter as the first parameter, but as the second parameter `replaceOne` expects a document with which it will replace the document matching the filter.

For example, in order to replace a document:

```
{
  "_id" : ObjectId("4b2b9f67a1f631733d917a7a"),
  "name" : "alice",
  "friends" : 24,
  "enemies" : 2
}
```

118. How do you Delete a Document?

The CRUD API in MongoDB provides `deleteOne` and `deleteMany` for this purpose. Both of these methods take a filter document as their first parameter. The filter specifies a set of criteria to match against in removing documents.

For example:

```
> db.books.deleteOne({"_id" : 3})
```

119. How to perform queries in MongoDB?

The `find` method is used to perform queries in MongoDB. Querying returns a subset of documents in a collection, from no documents at all to the entire collection. Which documents get returned is determined by the first argument to `find`, which is a document specifying the query criteria.

Example:

```
> db.users.find({"age" : 24})
```

120. What are the data types in MongoDB?

MongoDB supports a wide range of data types as values in documents. Documents in MongoDB are similar to objects in JavaScript. Along with JSON's essential key/value-pair nature, MongoDB adds support for a number of additional data types. The common data types in MongoDB are:

- Null
`{"x" : null}`
- Boolean
`{"x" : true}`
- Number
`{"x" : 4}`
- String
`{"x" : "foobar"}`
- Date
`{"x" : new Date() }`
- Regular expression
`{"x" : /foobar/i}`
- Array
`{"x" : ["a", "b", "c"]}`
- Embedded document
`{"x" : {"foo" : "bar"}}`
- Object ID
`{"x" : ObjectId() }`
- Binary Data
Binary data is a string of arbitrary bytes.
- Code
`{"x" : function() { /* ... */ } }`

121. When to use MongoDB?

You should use MongoDB when you are building internet and business applications that need to evolve quickly and scale elegantly. MongoDB is popular with developers of all kinds who are building scalable applications using agile methodologies.

MongoDB is a great choice if one needs to:

- Support a rapid iterative development.
- Scale to high levels of read and write traffic - MongoDB supports horizontal scaling through Sharding, distributing data across several machines, and facilitating high throughput operations with large sets of data.
- Scale your data repository to a massive size.
- Evolve the type of deployment as the business changes.
- Store, manage and search data with text, geospatial, or time-series dimensions.

122. How is Querying done in MongoDB?

The `find` method is used to perform queries in MongoDB. Querying returns a subset of documents in a collection, from no documents at all to the entire collection. Which documents get returned is determined by the first argument to `find`, which is a document specifying the query criteria.

For example: If we have a string we want to match, such as a "username" key with the value "alice", we use that key/value pair instead:

```
> db.users.find({"username" : "alice"})
```

123. Explain the term “Indexing” in MongoDB.

In MongoDB, indexes help in efficiently resolving queries. What an Index does is that it stores a small part of the data set in a form that is easy to traverse. The index stores the value of the specific field or set of fields, ordered by the value of the field as specified in the index.

MongoDB’s indexes work almost identically to typical relational database indexes.

Indexes look at an ordered list with references to the content. These in turn allow MongoDB to query orders of magnitude faster. To create an index, use the `createIndex` collection method.

For example:

```
> db.users.find({"username": "user101"}).explain("executionStats")
```

Here, `executionStats` mode helps us understand the effect of using an index to satisfy queries.

124. What are Geospatial Indexes in MongoDB?

MongoDB has two types of geospatial indexes: `2dsphere` and `2d`. `2dsphere` indexes work with spherical geometries that model the surface of the earth based on the WGS84 datum. This datum model the surface of the earth as an oblate spheroid, meaning that there is some flattening at the poles. Distance calculations using `2sphere` indexes, therefore, take the shape of the earth into account and provide a more accurate treatment of distance between, for example, two cities, than do `2d` indexes. Use `2d` indexes for points stored on a two-dimensional plane.

2dsphere allows you to specify geometries for points, lines, and polygons in the GeoJSON format. A point is given by a two-element array, representing [longitude, latitude]:

```
{
  "name" : "New York City",
  "loc" : {
    "type" : "Point",
    "coordinates" : [50, 2]
  }
}
```

A line is given by an array of points:

```
{
  "name" : "Hudson River",
  "loc" : {
    "type" : "LineString",
    "coordinates" : [[0,1], [0,2], [1,2]]
  }
}
```

125. Explain the SET Modifier in MongoDB?

If the value of a field does not yet exist, the "\$set" sets the value. This can be useful for updating schemas or adding user-defined keys.

Example:

```
> db.users.findOne()
{
  "_id" : ObjectId("4b253b067525f35f94b60a31"),
  "name" : "alice",
  "age" : 23,
  "sex" : "female",
  "location" : "India"
}
```

To add a field to this, we use "\$set":

```
> db.users.updateOne({"_id" :
ObjectId("4b253b067525f35f94b60a31")},
... {"$set" : {"favorite book" : "Start with Why"}})
```

126. What do you mean by Transactions?

A transaction is a logical unit of processing in a database that includes one or more database operations, which can be read or write operations. Transactions provide a useful feature in MongoDB to ensure consistency.

MongoDB provides two APIs to use transactions.

- **Core API:** It is a similar syntax to relational databases (e.g., `start_transaction` and `commit_transaction`)
- **Call-back API:** This is the recommended approach to using transactions. It starts a transaction, executes the specified operations, and commits (or aborts on the error). It also automatically incorporates error handling logic for "TransientTransactionError" and "UnknownTransactionCommitResult".

127. What are MongoDB Charts?

MongoDB Charts is a new, integrated tool in MongoDB for data visualization.

MongoDB Charts offers the best way to create visualizations using data from a MongoDB database.

It allows users to perform quick data representation from a database without writing code in a programming language such as Java or Python.

The two different implementations of MongoDB Charts are:

- MongoDB Charts PaaS (Platform as a Service)
- MongoDB Charts Server

128. What is the Aggregation Framework in MongoDB?

- The aggregation framework is a set of analytics tools within MongoDB that allow you to do analytics on documents in one or more collections.
- The aggregation framework is based on the concept of a pipeline. With an aggregation pipeline, we take input from a MongoDB collection and pass the documents from that collection through one or more stages, each of which performs a different operation on its inputs (See figure below). Each stage takes as input whatever the stage before it produced as output. The inputs and outputs for all stages are documents—a stream of documents.

129. What is a Replica Set in MongoDB?

To keep identical copies of your data on multiple servers, we use replication. It is recommended for all production deployments. Use replication to keep your application running and your data safe, even if something happens to one or more of your servers.

Such replication can be created by a replica set with MongoDB. A replica set is a group of servers with one primary, the server taking writes, and multiple secondaries, servers that keep copies of the primary's data. If the primary crashes, the secondaries can elect a new primary from amongst themselves.

130. What are some utilities for backup and restore in MongoDB?

The mongo shell does not include functions for exporting, importing, backup, or restore. However, MongoDB has created methods for accomplishing this, so that no scripting work or complex GUIs are needed. For this, several utility scripts are provided that can be used to get data in or out of the database in bulk. These utility scripts are:

- mongoimport
- mongoexport
- mongodump
- mongorestore