

## FAMILIARIZATION OF SQL COMMANDS

### Data Definition Language (DDL) Statements

#### 1. Create

To create tables, views, synonyms, sequences, functions, procedures, packages etc.

`CREATE TABLE <tablename>( attribute1 datatype, attribute2 datatype,... attribute n datatype);`

#### 2. Alter

To alter the structure of a table.

➤ To add new columns

`ALTER TABLE <tablename> add (attribute1 datatype);`

➤ To modify a column.

`ALTER TABLE <tablename> modify (attribute datatype);`

➤ To drop columns

`ALTER TABLE <tablename> drop column <column name>;`

#### 3. Rename

To rename a table, view, sequence, or private synonym for a table, view, or sequence.

Oracle automatically transfers integrity constraints, indexes, and grants on the old object to the new object.

`RENAME <oldtablename> to <newtablename>`

#### **4. Drop**

To permanently delete the values and the structure of tables, functions, procedures, views, synonym, sequences etc.

DROP table <tablename>;

#### **5. Truncate**

Use the Truncate statement to delete all the rows from table permanently.

TRUNCATE table <tablename>;

### **Data Manipulation Language (DML) Statements**

#### **1. Insert**

Used to add new rows to a table.

INSERT INTO <table name>VALUES (<value 1>, ... <value n>);

#### **2. Update**

The update statement is used to change values that are already in a table.

UPDATE <table name> SET <attribute> = <expression> WHERE <condition>;

#### **3. Delete**

To delete a particular row from the table based on a condition.

**DELETE FROM <table name> WHERE <condition>;**

If the WHERE clause is omitted, then every row of the table is deleted.

#### **4. Select**

A SELECT statement can include the following:

1. The name of each column you want to include
2. The name of the table or view that contains the data
3. A search condition to uniquely identify the row that contains the information you want
4. The name of each column used to group your data
5. A search condition that uniquely identifies a group that contains the information you want
6. The order of the results so a specific row among duplicates can be returned.

**SELECT** column names

**FROM** table or view name

**WHERE** search condition

**GROUP BY** column names

**HAVING** search condition

**ORDER BY** column-name

The SELECT and FROM clauses must be specified. The other clauses are optional.

To retrieve all columns (in the same order as they appear in the table's definition), use an asterisk (\*) instead of naming the columns:

**SELECT \*FROM <TABLENAME>;**

The FROM clause specifies the table that you want to select data from. You can select columns from more than one table.

## **Transaction Control Statements (TCL)**

TCL statements are used to manage the changes made by DML statements. A transaction is a set of SQL statements which Oracle treats as a Single Unit i.e. all the statements should execute successfully or none of the statements should execute.

### **1. Commit**

To make the changes done in a transaction permanent, issue the COMMIT statement.

### **2. Rollback**

To rollback the changes done in a transaction, give ROLLBACK statement.

Rollback restores the state of the database to the last commit point.

### **3. Savepoint**

To specify a point in a transaction to which later you can roll back.

```
SAVEPOINT savepointname;
```

## **Data Control Language (DCL) Statements**

DCL statements are used to control access to data stored in a database.

### **1. Grant**

To provide privileges and accesses on database objects to the users.

```
GRANT <privilege name> ON <object name> TO <user name/role name/public>;
```

## **2. Revoke**

To remove the previously granted privileges from the user.

## VIVA QUESTIONS

### CYCLE I

#### 1. What is SQL?

Structured Query Language

#### 2. 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.

#### 3. 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.

#### 4. What is a Database system?

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

#### 5. Advantages of DBMS?

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

#### 6. Disadvantage in File Processing System?

- Data redundancy & inconsistency.
- Difficult in accessing data.
- Data isolation.

- Data integrity.
- Concurrent access is not possible.
- Security Problems.

**7. Describe the three levels of data abstraction?**

The are three levels of abstraction:

- *Physical level*: The lowest level of abstraction describes how data are stored.
- *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.

**8 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.

**9 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.

**10. 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.

**11 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

**12 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:

- Physical Data Independence: Modification in physical level should not affect the logical level.
- Logical Data Independence: Modification in logical level should affect the view level.

*NOTE: Logical Data Independence is more difficult to achieve*

### **13 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 direct 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.

### **14 What is Data Model?**

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

### **15 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.

### **16 What is Object Oriented model?**

This model is based on collection of objects. An object contains values stored in instance variables with in 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.

**25. What is DDL (Data Definition Language)?**

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

**26. What is VDL (View Definition Language)?**

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

**27. What is SDL (Storage Definition Language)?**

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

**28. 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.

**29. What is DML (Data Manipulation Language)?**

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

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

**31. What is DML Compiler?**

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

## CYCLE II

31. What is PL-SQL ?

Procedural Language Structured Query Language

32. What is Query evaluation engine?

It executes low-level instruction generated by compiler.

33. What is DDL Interpreter?

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

34. 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.

35. 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.

36. 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.

37. 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.

### **38. 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.

### **39. 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

- Minimizing redundancy
- Minimizing insertion, deletion and update anomalies.

### **40. 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 t1 and t2 in r if  $t1[X] = t2[X]$  then they have  $t1[Y] = t2[Y]$ . This means the value of X component of a tuple uniquely determines the value of component Y.

### **41. When is a functional dependency F said to be minimal?**

- Every dependency in F has a single attribute for its right hand side.
- 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.
- We cannot remove any dependency from F and still have set of dependency that is equivalent to F.

### **42. What is Multivalued dependency?**

Multivalued dependency denoted by  $X \multimap Y$  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 t1 and t2 exist in r such that  $t1[X] = t2[X]$  then t3 and t4 should also exist in r with the following properties

- $t3[X] = t4[X] = t1[X] = t2[X]$
- $t3[Y] = t1[Y]$  and  $t4[Y] = t2[Y]$
- $t3[Z] = t2[Z]$  and  $t4[Z] = t1[Z]$   
where  $[Z = (R - (X \cup Y))]$

### **43. What is Lossless join property?**

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

---

## CYCLE III

### 44. What is 1 NF (Normal Form)?

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

### 45. What is Fully Functional dependency?

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

### 46. 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.

### 47. What is 3NF?

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

- X is a Super-key of R.
- A is a prime attribute of R.

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

### 48. 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  $X \rightarrow A$ , X must be a candidate key.

### 49. What is 4NF?

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

- X is subset or equal to (or)  $XY = R$ .
- X is a super key.

## ADDITIONAL QUESTIONS

**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?**

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

**5. Disadvantage in File Processing System?**

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

**6. Describe the three levels of data abstraction?**

The are three levels of abstraction:

- Physical level: The lowest level of abstraction describes how data are stored.

- 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.

1. Entity Integrity: States that "Primary key cannot have NULL value"
2. 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:

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

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 account 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 instances.

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 a 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 object.

**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 a 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 and the list of attributes  $A_i$  that it contains. A relation is defined as a set of tuples. Let  $r$  be a relation which contains set tuples  $(t_1, t_2, t_3, \dots, t_n)$ . Each tuple is an ordered list of n-values  $(v_1, v_2, \dots, v_n)$ .

**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 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 specified 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 SCL (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

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 model.

- Procedural DML or Low level: DML requires a user to specify what data needed and how to get those data.
- Non-Procedural DML or High level: DML requires a user to specify what data 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 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.

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 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 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

- Minimizing redundancy
- 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  $Y$ . If  $X$  are subsets of  $R$  specifies a constraint on the possible tuple that can form a relation structure 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?

- Every dependency in  $F$  has a single attribute for its right hand side.
- We cannot replace any dependency  $X A$  in  $F$  with a dependency  $Y A$  where  $Y$  is a proper subset of  $X$  and still have a set of dependency that is equivalent to  $F$ .

- 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  $X \rightarrow Y$  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 t1 and t2 exist in r such that  $t1[X] = t2[X]$  then t3 and t4 should also exist in r with the following properties

$$t3[X] = t4[X] = t1[X] = t2[X]$$

$$t3[Y] = t1[Y] \text{ and } t4[Y] = t2[Y]$$

$$t3[Z] = t2[Z] \text{ and } t4[Z] = t1[Z]$$

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  $X \rightarrow Y$  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 A either of the following is true

- X is a Super-key of R.
- 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 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 Y that holds over R, one of following is true

- X is subset or equal to (or) XY = R.
- 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

- Ri = R for some i.
- 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:**

- Binary search style indexing
- B-Tree indexing
- Inverted list indexing
- Memory resident table
- 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 {R1, R2, ..., Rn} is said to hold over a relation R if R1, R2, R3, ..., Rn 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

- Analysis phase
- Redo Phase
- 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 database. 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 tablespace 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. 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

- Database files
- Control files
- 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.

**78. 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

**79. 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).

**80. 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.

## 81. 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.

- Oracle database file number, which contains the block with the rows
- Oracle block address, which contains the row
- 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

## 82. 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.

**83. Name two utilities that Oracle provides, which are use 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.

**84. 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.

**85. 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.

**86. 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.

**87. 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.

**88. 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.

**89. 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.

**90. 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.

**91. 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

- Implicit
- Explicit

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

The well-known inference rules for FDs

- **Reflexive rule :**  
If Y is subset or equal to X then X Y.
- **Augmentation rule:**  
If X Y then XZ YZ.
- **Transitive rule:**  
If {X Y, Y Z} then X Z.

- Decomposition rule :  
If  $X \rightarrow YZ$  then  $X \rightarrow Y$ .
- Union or Additive rule:  
If  $\{X \rightarrow Y, X \rightarrow Z\}$  then  $X \rightarrow YZ$ .
- Pseudo Transitive rule :  
If  $\{X \rightarrow Y, WY \rightarrow Z\}$  then  $X \rightarrow WZ$ .

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.

**93. 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.

**94. 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 .