INTRODUCTION TO SQL

Pronounced as SEQUEL: Structured English QUERY Language

- Pure non-procedural query language
- Designed and developed by IBM, Implemented by Oracle
- 1978 System/R IBM- 1st Relational DBMS
- 1979 Oracle and Ingres
- 1982 SQL/DS and DB2 IBM
- Accepted by both ANSI + ISO as **Standard Query Language** for any RDBMS
- SQL86 (SQL1) : first by ANSI and ratified by ISO (SQL-87), minor revision on 89 (SQL-89)
- SQL92 (SQL2) : major revision
- SQL99 (SQL3) : add recursive query, trigger, some OO features, and non-scholar type
- SQL2003 : XML, Window functions, and sequences (Not free)
- Supports all the three sublanguages of DBMS: **DDL, DML, DCL**
- Supports Aggregate functions, String Manipulation functions, Set theory operations, Date Manipulation functions, rich set of operators (IN, BETWEEN, LIKE, IS NULL, EXISTS)
- Supports REPORT writing features and Forms for designing GUI based applications

DATA DEFINITION, CONSTRAINTS, AND SCHEMA CHANGES

Used to CREATE, ALTER, and DROP the descriptions of the database tables (relations)

Data Definition in SQL

CREATE, ALTER and DROP

table……..……relation

row……..…….tuple

column……..……attribute

DATA TYPES

- Numeric: NUMBER, NUMBER(s,p), INTEGER, INT, FLOAT, DECIMAL
- Character: CHAR(n), VARCHAR(n), VARCHAR2(n), CHAR VARYING(n)
- Bit String: BLOB, CLOB
- Boolean: true, false, and null

- Date and Time: DATE (YYYY-MM-DD) TIME(HH:MM:SS)
- Timestamp: DATE + TIME
- USER Defined types

CREATE SCHEMA

Specifies a new database schema by giving it a name

Ex: CREATE SCHEMA COMPANY AUTHORIZATION Jsmith;

CREATE TABLE

- Specifies a new base relation by giving it a name, and specifying each of its attributes and their data types

  Syntax of CREATE Command:

  CREATE TABLE <*table name***>** (<Attribute *A*1> <Data Type *D*1> [< Constraints>],

  <Attribute *A*2> <Data Type *D*2> [< Constraints>],

  …….

  <Attribute *A*n> <Data Type *D*n> [< Constraints>],

  [<integrity-constraint1>, <integrity-constraint k>]);

  - A constraint NOT NULL may be specified on an

  attribute A constraint NOT NULL may be specified on an

  attribute Ex: CREATE TABLE DEPARTMENT (

  DNAME VARCHAR(10) NOT NULL,

  DNUMBER INTEGER NOT NULL,

  MGRSSN CHAR(9), MGRSTARTDATE CHAR(9));

- Specifying the unique, primary key attributes, secondary keys, and referential integrity constraints (foreign keys).

  Ex: CREATE TABLE DEPT (DNAME

  VARCHAR(10) NOT NULL,

  DNUMBER INTEGER NOT NULL,

  MGRSSN CHAR(9),

  MGRSTARTDATE CHAR(9),

  PRIMARY KEY (DNUMBER),

  UNIQUE (DNAME),

FOREIGN KEY (MGRSSN) REFERENCES EMP(SSN));

- We can specify RESTRICT, CASCADE, SET NULL or SET DEFAULT on referential integrity constraints (foreign keys)

  Ex: CREATE TABLE DEPT (DNAME
  VARCHAR(10) NOT NULL,
  DNUMBER INTEGER NOT NULL,
  MGRSSN CHAR(9), MGRSTARTDATE CHAR(9),
  PRIMARY KEY (DNUMBER),
  UNIQUE (DNAME),
  FOREIGN KEY (MGRSSN) REFERENCES EMP
  ON DELETE SET DEFAULT ON UPDATE CASCADE);

DROP TABLE

- Used to remove a relation (base table) and its definition.
- The relation can no longer be used in queries, updates, or any other commands since its description no longer exists

  Example: DROP TABLE DEPENDENT;

ALTER TABLE:

- Used to add an attribute to/from one of the base relations drop constraint -- The new attribute will have NULLs in all the tuples of the relation right after the command is executed; hence, the NOT NULL constraint is *not allowed* for such an attribute.
  Example: ALTER TABLE EMPLOYEE ADD JOB VARCHAR2 (12);
- The database users must still enter a value for the new attribute JOB for each EMPLOYEE tuple. This can be done using the UPDATE command.

DROP A COLUMN (AN ATTRIBUTE)

- ALTER TABLE COMPANY.EMPLOYEE DROP ADDRESS CASCADE; All constraints and views that reference the column are dropped automatically, along with the column. ALTER TABLE COMPANY.EMPLOYEE DROP ADDRESS RESTRICT; Successful if no views or constraints reference the column. ALTER TABLE COMPANY.DEPARTMENT ALTER MGRSSN DROP DEFAULT;

- ALTER TABLE COMPANY.DEPARTMENT ALTER MGRSSN SET DEFAULT ―333445555‖;

BASIC QUERIES IN SQL

- SQL has one basic statement for retrieving information from a database; the SLELECT statement
- This is *not the same as* the SELECT operation of the relational algebra
- Important distinction between SQL and the formal relational model;
- SQL allows a table (relation) to have two or more tuples that are identical in all their attribute values
- Hence, an SQL relation (table) is a *multi-set* (sometimes called a bag) of tuples; it is *not* a set of tuples
- SQL relations can be constrained to be sets by using the CREATE UNIQUE INDEX command, or by using the DISTINCT option
- Basic form of the SQL SELECT statement is called a *mapping* of a *SELECT-FROM-WHERE block*

  SELECT <attribute list> FROM <table list> WHERE <condition>
- <attribute list> is a list of attribute names whose values are to be retrieved by the query
- <table list > is a list of the relation names required to process the query
- <condition> is a conditional (Boolean) expression that identifies the tuples to be retrieved by the query

SIMPLE SQL QUERIES

Basic SQL queries correspond to using the following operations of the relational algebra:

SELECT

PROJECT

JOIN

All subsequent examples uses COMPANY database as shown below:

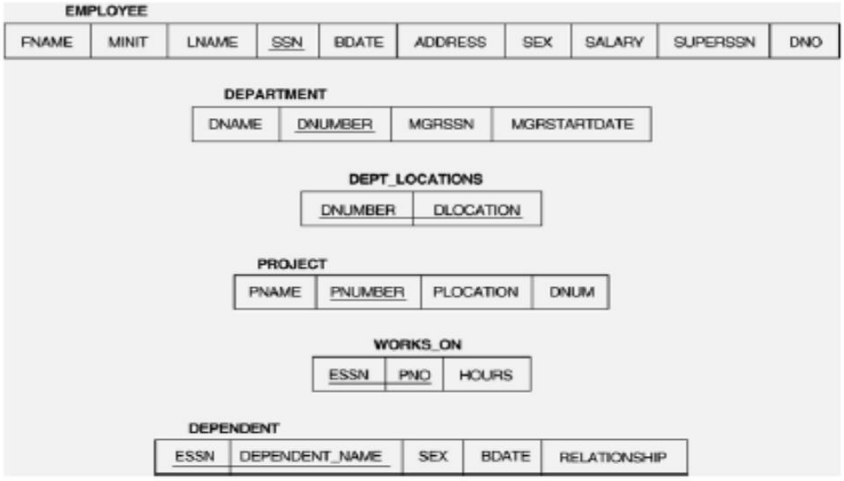
Example of a simple query on one relation

Query 0: Retrieve the birth date and address of the employee whose name is 'John B. Smith'.

Q0: SELECT BDATE, ADDRESS FROM EMPLOYEE

WHERE FNAME='John' AND MINIT='B' AND LNAME='Smith'

Similar to a SELECT-PROJECT pair of relational algebra operations: The SELECT-clause specifies the projection attributes and the WHERE-clause specifies the selection condition However, the result of the query may contain duplicate tuples



EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

DEPT_LOCATIONS	DNUMBER	DLOCATION
	1	Houston
	4	Stafford
	5	Bellaire
	5	Sugarland
	5	Houston

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	1988-05-22
	Administration	4	987654321	1995-01-01
	Headquarters	1	888665555	1981-06-19

WORKS_ON	ESSN	PNO	HOURS
	123456789	1	32.5
	123456789	2	7.5
	666884444	3	40.0
	453453453	1	20.0
	453453453	2	20.0
	333445555	2	10.0
	333445555	3	10.0
	333445555	10	10.0
	333445555	20	10.0
	999887777	30	30.0
	999887777	10	10.0
	987987987	10	35.0
	987987987	30	5.0
	987654321	30	20.0
	987654321	20	15.0
	888665555	20	null

PROJECT	PNAME	PNUMBER	PLOCATION	DNUM
	ProductX	1	Bellaire	5
	ProductY	2	Sugarland	5
	ProductZ	3	Houston	5
	Computerization	10	Stafford	4
	Reorganization	20	Houston	1
	Newbenefits	30	Stafford	4

DEPENDENT	ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
	333445555	Alice	F	1986-04-05	DAUGHTER
	333445555	Theodore	M	1983-10-25	SON
	333445555	Joy	F	1958-05-03	SPOUSE
	987654321	Abner	M	1942-02-28	SPOUSE
	123456789	Michael	M	1988-01-04	SON
	123456789	Alice	F	1988-12-30	DAUGHTER
	123456789	Elizabeth	F	1967-05-05	SPOUSE

Example of a simple query on two relations

Query 1: Retrieve the name and address of all employees who work for the 'Research' department.

Q1: SELECT FNAME, LNAME, ADDRESS FROM EMPLOYEE, DEPARTMENT WHERE DNAME='Research' AND DNUMBER=DNO

Similar to a SELECT-PROJECT-JOIN sequence of relational algebra operations (DNAME='Research') is a selection condition (corresponds to a SELECT operation in relational algebra) (DNUMBER=DNO) is a join condition (corresponds to a JOIN operation in relational algebra)

Example of a simple query on three relations

Query 2: For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birth date.

Q2: SELECT PNUMBER, DNUM, LNAME, BDATE, ADDRESS FROM PROJECT, DEPARTMENT, EMPLOYEE WHERE DNUM=DNUMBER AND MGRSSN=SSN AND PLOCATION='Stafford'

In Q2, there are two join conditions The join condition DNUM=DNUMBER relates a project to its controlling department The join condition MGRSSN=SSN relates the controlling department to the employee who manages that department

ALIASES, * AND DISTINCT, EMPTY WHERE-CLAUSE

- In SQL, we can use the same name for two (or more) attributes as long as the attributes are in different relations
- A query that refers to two or more attributes with the same name must qualify the attribute name with the relation name by prefixing the relation name to the attribute name **Example:** EMPLOYEE.LNAME, DEPARTMENT.DNAME
- Some queries need to refer to the same relation twice. In this case, aliases are given to the relation name

Example

Query 3: For each employee, retrieve the employee's name, and the name of his or her immediate supervisor.

Q3: SELECT E.FNAME, E.LNAME, S.FNAME, S.LNAME FROM EMPLOYEE E S WHERE E.SUPERSSN=S.SSN

In Q3, the alternate relation names E and S are called aliases or tuple variables for the EMPLOYEE relation We can think of E and S as two different copies of EMPLOYEE; E represents employees in role of supervisees and S represents employees in role of supervisors

Aliasing can also be used in any SQL query for convenience. Can also use the AS keyword to specify aliases

Q3: SELECT E.FNAME, E.LNAME, S.FNAME, S.LNAME FROM EMPLOYEE AS E, EMPLOYEE AS S WHERE E.SUPERSSN=S.SSN

UNSPECIFIED WHERE-clause

A missing WHERE-clause indicates no condition; hence, all tuples of the relations in the FROM-clause are selected. This is equivalent to the condition WHERE TRUE Example:

Query 4: Retrieve the SSN values for all employees.

Q4: SELECT SSN FROM EMPLOYEE

If more than one relation is specified in the FROM-clause and there is no join condition, then the CARTESIAN PRODUCT of tuples is selected

Example:

Q5: SELECT SSN, DNAME FROM EMPLOYEE, DEPARTMENT

Note: It is extremely important not to overlook specifying any selection and join conditions in the WHERE-clause; otherwise, incorrect and very large relations may result

USE OF *

To retrieve all the attribute values of the selected tuples, a * is used, which stands for all the attributes

Examples:

R**etrieve all the attribute values of EMPLOYEES who work in department 5.**

Q1a: SELECT * FROM EMPLOYEE WHERE DNO=5

Retrieve all the attributes of an employee and attributes of DEPARTMENT he works in for every employee of 'Research' department.

Q1b: SELECT * FROM EMPLOYEE, DEPARTMENT WHERE DNAME='Research' AND DNO=DNUMBER

USE OF DISTINCT

SQL does not treat a relation as a set; duplicate tuples can appear. To eliminate duplicate tuples in a query result, the keyword DISTINCT is used

Example: the result of **Q1c** may have duplicate SALARY values whereas **Q1d** does not have any duplicate values

Q1c: SELECT SALARY FROM EMPLOYEE Q1d: SELECT **DISTINCT** SALARY FROM EMPLOYEE

SET OPERATIONS

SQL has directly incorporated some set operations such as union operation (UNION), set difference (MINUS) and intersection (INTERSECT) operations. The resulting relations of these set operations are sets of tuples; duplicate tuples are eliminated from the result. The set operations apply only to union compatible relations; the two relations must have the same attributes and the attributes must appear in the same order

Query 5: Make a list of all project numbers for projects that involve an employee whose last name is 'Smith' as a worker or as a manager of the department that controls the project.

Q5: (SELECT PNAME FROM PROJECT, DEPARTMENT, EMPLOYEE WHERE DNUM=DNUMBER AND MGRSSN=SSN AND LNAME='Smith')

UNION

(SELECT PNAME FROM PROJECT, WORKS_ON, EMPLOYEE WHERE PNUMBER=PNO AND ESSN=SSN AND NAME='Smith')

NESTING OF QUERIES

A complete SELECT query, called a nested query, can be specified within the WHERE-clause of another query, called the outer query. Many of the previous queries can be specified in an alternative form using nesting

Query 6: Retrieve the name and address of all employees who work for the 'Research' department.

Q6: SELECT FNAME, LNAME, ADDRESS FROM EMPLOYEE WHERE DNO **IN**
(SELECT DNUMBER FROM DEPARTMENT WHERE DNAME='Research')

Note: The nested query selects the number of the 'Research' department. The outer query selects an EMPLOYEE tuple if its DNO value is in the result of either nested query. The comparison operator IN compares a value v with a set (or multi-set) of values V, and evaluates to TRUE if v is one of the elements in V

In general, we can have several levels of nested queries. A reference to an unqualified attribute refers to the relation declared in the innermost nested query. In this example, the nested query is not correlated with the outer query

CORRELATED NESTED QUERIES

If a condition in the WHERE-clause of a nested query references an attribute of a relation declared in the outer query, the two queries are said to be correlated. The result of a correlated nested query is different for each tuple (or combination of tuples) of the relation(s) the outer query

Query 7: Retrieve the name of each employee who has a dependent with the same first name as the employee.

Q7: SELECT E.FNAME, E.LNAME FROM EMPLOYEE AS E WHERE E.SSN **IN**
(SELECT ESSN FROM DEPENDENT WHERE ESSN=E.SSN AND E.FNAME=DEPENDENT_NAME)

In Q7, the nested query has a different result in the outer query. A query written with nested SELECT... FROM… WHERE... blocks and using the **= or IN** comparison operators can *always* be expressed as a single block query. For example, Q7 may be written as in Q7a

Q7a: SELECT E.FNAME, E.LNAME FROM EMPLOYEE E, DEPENDENT D
WHERE E.SSN=D.ESSN AND E.FNAME=D.DEPENDENT_NAME

THE EXISTS FUNCTION

EXISTS is used to check whether the result of a correlated nested query is empty (contains no tuples) or not. We can formulate Query 7 in an alternative form that uses EXIST.

Q7b: SELECT FNAME, LNAME FROM EMPLOYEE
WHERE **EXISTS** (SELECT * FROM DEPENDENT WHERE SSN=ESSN
AND FNAME=DEPENDENT_NAME)

Query 8: Retrieve the names of employees who have no dependents.

Q8: SELECT FNAME, LNAME FROM EMPLOYEE

WHERE **NOT EXISTS**

(SELECT * FROM DEPENDENT WHERE SSN=ESSN)

Note: In Q8, the correlated nested query retrieves all DEPENDENT tuples related to an EMPLOYEE tuple. If none exist, the EMPLOYEE tuple is selected

EXPLICIT SETS

It is also possible to use an explicit (enumerated) set of values in the WHERE-clause rather than a nested query

Query 9: Retrieve the social security numbers of all employees who work on project number 1, 2, or 3.

Q9: SELECT DISTINCT ESSN FROM WORKS_ON WHERE PNO **IN (1, 2, 3)**

NULLS IN SQL QUERIES

SQL allows queries that check if a value is NULL (missing or undefined or not applicable). SQL uses IS or IS NOT to compare NULLs because it considers each NULL value distinct from other NULL values, so equality comparison is not appropriate.

Query 10: Retrieve the names of all employees who do not have supervisors.

Q10: SELECT FNAME, LNAME FROM EMPLOYEE

WHERE SUPERSSN IS NULL

Note: If a join condition is specified, tuples with NULL values for the join attributes are not included in the result

AGGREGATE FUNCTIONS

Include COUNT, SUM, MAX, MIN, and AVG

Query 11: Find the maximum salary, the minimum salary, and the average salary among all employees.

Q11: SELECT **MAX (SALARY), MIN(SALARY), AVG(SALARY)**

FROM EMPLOYEE

Note: Some SQL implementations may not allow more than one function in the SELECT-clause

Query 12: Find the maximum salary, the minimum salary, and the average salary among employees who work for the 'Research' department.

Q12: SELECT **MAX (SALARY), MIN(SALARY), AVG(SALARY)** FROM EMPLOYEE, DEPARTMENT WHERE DNO=DNUMBER AND DNAME='Research'

Queries 13 and 14: Retrieve the total number of employees in the company (Q13), and the number of employees in the 'Research' department (Q14).

Q13: SELECT **COUNT (*)** FROM EMPLOYEE

Q14: SELECT **COUNT (*)** FROM EMPLOYEE, DEPARTMENT

WHERE DNO=DNUMBER AND DNAME='Research‘

GROUPING

- In many cases, we want to apply the aggregate functions to subgroups of tuples in a relation
- Each subgroup of tuples consists of the set of tuples that have the same value for the grouping attribute(s)
- The function is applied to each subgroup independently
- SQL has a GROUP BY-clause for specifying the grouping attributes, which must also appear in the SELECT-clause

Query 15: For each department, retrieve the department number, the number of employees in the department, and their average salary.

Q15: SELECT DNO, COUNT (*), AVG (SALARY)

FROM EMPLOYEE **GROUP BY** DNO

- In Q15, the EMPLOYEE tuples are divided into groups. Each group having the same value for the grouping attribute DNO
- The COUNT and AVG functions are applied to each such group of tuples separately
- The SELECT-clause includes only the grouping attribute and the functions to be applied on each group of tuples
- A join condition can be used in conjunction with grouping

Query 16: For each project, retrieve the project number, project name, and the number of employees who work on that project.

Q16: SELECT PNUMBER, PNAME, COUNT (*)

```
FROM PROJECT, WORKS_ON
WHERE PNUMBER=PNO
GROUP BY PNUMBER, PNAME
```

THE HAVING-CLAUSE

Sometimes we want to retrieve the values of these functions for only those groups that satisfy certain conditions. The HAVING-clause is used for specifying a selection condition on groups (rather than on individual tuples)

Query 17: For each project on which more than two employees work, retrieve the project number, project name, and the number of employees who work on that project.

```
Q17: SELECT PNUMBER, PNAME, COUNT
(*) FROM PROJECT, WORKS_ON WHERE
PNUMBER=PNO GROUP BY PNUMBER,
PNAME
HAVING COUNT (*) > 2
```

SUBSTRING COMPARISON

The LIKE comparison operator is used to compare partial strings. Two reserved characters are used: **'%'** (or '*' in some implementations) replaces an arbitrary number of characters, and **'_'** replaces a single arbitrary character.

Query 18: Retrieve all employees whose address is in Houston, Texas. Here, the value of the ADDRESS attribute must contain the substring 'Houston,TX‘ in it.

```
Q18: SELECT FNAME, LNAME
FROM EMPLOYEE WHERE ADDRESS LIKE '%Houston,TX%'
```

Query 19: Retrieve all employees who were born during the 1950s.

Here, '5' must be the 8th character of the string (according to our format for date), so the BDATE value is '_____5_', with each underscore as a place holder for a single arbitrary character.

```
Q19: SELECT FNAME, LNAME
FROM EMPLOYEE WHERE BDATE LIKE '_____5_‘
```

Note: The LIKE operator allows us to get around the fact that each value is considered atomic and indivisible. Hence, in SQL, character string attribute values are not atomic

ARITHMETIC OPERATIONS

The standard arithmetic operators '+', '-'. '*', and '/' (for addition, subtraction, multiplication, and division, respectively) can be applied to numeric values in an SQL query result

Query 20: Show the effect of giving all employees who work on the 'ProductX' project a 10% raise.

```
Q20: SELECT FNAME, LNAME, 1.1*SALARY
FROM EMPLOYEE, WORKS_ON, PROJECT
WHERE SSN=ESSN
AND PNO=PNUMBER AND PNAME='ProductX‘
```

ORDER BY

The ORDER BY clause is used to sort the tuples in a query result based on the values of some attribute(s)

Query 21: Retrieve a list of employees and the projects each works in, ordered by the employee's department, and within each department ordered alphabetically by employee last name.

```
Q21: SELECT DNAME, LNAME, FNAME, PNAME
FROM DEPARTMENT, EMPLOYEE, WORKS_ON, PROJECT
WHERE DNUMBER=DNO
AND SSN=ESSN
AND PNO=PNUMBER
ORDER BY DNAME, LNAME
```

The default order is in ascending order of values. We can specify the keyword DESC if we want a descending order; the keyword ASC can be used to explicitly specify ascending order, even though it is the default

Ex: ORDER BY DNAME **DESC**, LNAME **ASC**, FNAME **ASC**

MORE EXAMPLE QUERIES:

Query 22: Retrieve the names of all employees who have two or more dependents.

```
Q22: SELECT LNAME, FNAME FROM
EMPLOYEE
```

```
WHERE (SELECT COUNT (*) FROM DEPENDENT
    WHERE SSN=ESSN) ≥ 2);
```

Query 23: List the names of managers who have least one dependent.

```
Q23: SELECT FNAME, LNAME
    FROM EMPLOYEE
 WHERE EXISTS (SELECT * FROM DEPENDENT WHERE SSN=ESSN)
AND EXISTS ( SELECT * FROM DEPARTMENT WHERE SSN=MGRSSN );
```

SPECIFYING UPDATES IN SQL

There are three SQL commands to modify the database: **INSERT**, **DELETE**, and **UPDATE.**

INSERT

- In its simplest form, it is used to add one or more tuples to a relation
- Attribute values should be listed in the same order as the attributes were specified in the **CREATE TABLE** command

Example:

```
INSERT INTO EMPLOYEE VALUES ('Richard','K','Marini', '653298653', '30-DEC-52',
'98 Oak Forest,Katy,TX', 'M', 37000,'987654321', 4 )
```

- An alternate form of INSERT specifies explicitly the attribute names that correspond to the values in the new tuple. Attributes with NULL values can be left out

Example: Insert a tuple for a new EMPLOYEE for whom we only know the FNAME, LNAME, and SSN attributes.

```
INSERT INTO EMPLOYEE (FNAME, LNAME, SSN)VALUES ('Richard', 'Marini',
'653298653')
```

Important Note: Only the constraints specified in the DDL commands are automatically enforced by the DBMS when updates are applied to the database. Another variation of INSERT allows insertion of multiple tuples resulting from a **query** into a relation

Example: Suppose we want to create a temporary table that has the name, number of employees, and total salaries for each department. A table DEPTS_INFO is created first, and is loaded with the summary information retrieved from the database by the query.

```
CREATE TABLE DEPTS_INFO
 (DEPT_NAME VARCHAR (10),
```

```
NO_OF_EMPS INTEGER, TOTAL_SAL INTEGER);

INSERT INTO DEPTS_INFO (DEPT_NAME, NO_OF_EMPS, TOTAL_SAL)
SELECT DNAME, COUNT (*), SUM (SALARY) FROM DEPARTMENT,
EMPLOYEE WHERE DNUMBER=DNO GROUP BY DNAME ;
```

Note: The DEPTS_INFO table may not be up-to-date if we change the tuples in either the DEPARTMENT or the EMPLOYEE relations *after* issuing the above. We have to create a view (see later) to keep such a table up to date.

DELETE

- Removes tuples from a relation. Includes a WHERE-clause to select the tuples to be deleted
- Referential integrity should be enforced
- Tuples are deleted from only *one table* at a time (unless CASCADE is specified on a referential integrity constraint)
- A missing WHERE-clause specifies that *all tuples* in the relation are to be deleted; the table then becomes an empty table
- The number of tuples deleted depends on the number of tuples in the relation that satisfy the WHERE-clause

Examples:

```
1: DELETE FROM EMPLOYEE WHERE LNAME='Brown‘;
2: DELETE FROM EMPLOYEE WHERE SSN='123456789‘;
3:  DELETE FROM EMPLOYEE WHERE DNO IN (SELECT DNUMBER
FROM DEPARTMENT WHERE DNAME='Research');
4: DELETE FROM EMPLOYEE;
```

UPDATE

- Used to modify attribute values of one or more selected tuples
- A WHERE-clause selects the tuples to be modified
- An additional SET-clause specifies the attributes to be modified and their new values
- Each command modifies tuples *in the same relation*
- Referential integrity should be enforced

Example1: Change the location and controlling department number of project number 10 to 'Bellaire' and 5, respectively.

```
UPDATE PROJECT
SET PLOCATION = 'Bellaire', DNUM = 5 WHERE PNUMBER=10;
```

Example2: Give all employees in the 'Research' department a 10% raise in salary.

```
UPDATE EMPLOYEE
SET SALARY = SALARY *1.1
WHERE DNO IN (SELECT DNUMBER FROM DEPARTMENT
WHERE DNAME='Research');
```

SQL TRIGGERS

- Objective: to monitor a database and take initiate action when a condition occurs
- Triggers are nothing but the procedures/functions that involve actions and fired/executed automatically whenever an event occurs such as an insert, delete, or update operation or pressing a button or when mouse button is clicked

VIEWS IN SQL

- A view is a single *virtual table* that is derived from other tables. The other tables could be base tables or previously defined view.
- Allows for limited update operations Since the table may not physically be stored
- Allows full query operations
- A convenience for expressing certain operations
- A view does not necessarily exist in physical form, which limits the possible update operations that can be applied to views.

Viva Questions

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.
- Unauthorized 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?

There 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 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

11. 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.

12. What is Data Model?

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

13. 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.

14. 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.

15. What is an Entity?

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

16. What is an Entity type?

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

17. What is an Entity set?

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

18. What is an Extension of entity type?

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

19. What is an attribute?

It is a particular property, which describes the entity.

20. What is a Relation Schema and a Relation?

A relation Schema denoted by R(A1, A2, …, An) 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 (t1, t2, t3, ...,tn). Each tuple is an ordered list of n-values t=(v1,v2, ..., vn).

21. What is degree of a Relation?

It is the number of attribute of its relation schema.

22. What is Relationship?

It is an association among two or more entities.

23. What is Relationship set?

The collection (or set) of similar relationships.

24. *What is Relationship type?*

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

25. What is degree of Relationship type?

It is the number of entity type participating.

26. What is DDL (Data Definition Language)?

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

27. What is VDL (View Definition Language)?

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

28. What is SDL (Storage Definition Language)?

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

29. 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.

30. What is DML (Data Manipulation Language)?

This language that enable user to access or manipulate data as organized 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.

32. What is Relational Algebra?

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

33. 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.

34. What is normalization?

It is a process of analyzing 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.

35. What is Functional Dependency?

A Functional dependency is denoted by X Y between 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.

36. 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 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.

37. What is Multivalued dependency?

Multivalued dependency denoted by X → 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] and t4[Y] = t2[Y]
- t3[Z] = t2[Z] and t4[Z] = t1[Z]
  where [Z = (R-(X U Y))]

38. What is Lossless join property?

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

39. What is 1 NF (Normal Form)?

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

40. What is Fully Functional dependency?

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

41. 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.

42. 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.

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

A relation schema R is in BCNF if it is in 3NF and satisfies additional constraints that for every FD X→A, X must be a candidate key.

44. 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.

45. 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.

46. What is Domain-Key Normal Form?

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

47. 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.

ArtificialKey:

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.

CompoundKey**:**

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.

NaturalKey**:**

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

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

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

- Binary search style indexing
- B-Tree indexing
- Inverted list indexing
- Memory resident table
- Table indexing

49. 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.

50. 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.

51. 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.

52. 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.

53. 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.

54. 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.

55. 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.

56. What are the different phases of transaction?

Different phases are

- Analysis phase
- Redo Phase
- Undo phase

57. 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.

58. What is "transparent DBMS"?

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

59. 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.

60. 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.

61. What do you mean by Correlated subquery?

Sub queries, 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)

62. What are the primitive operations common to all record management systems? Addition, deletion and modification.

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

64. What are the unary operations in Relational Algebra?

PROJECTION and SELECTION.

65. 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.

66. 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.

67. 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

68. 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.

69. 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.

70. Not only RDBMS takes care of locating data it also Determines

an optimal access path to store or retrieve the data

71. How do you communicate with an RDBMS?

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

72. 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.

73. 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.

74. 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

75. 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).

76. 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 redoes 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.

77. 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

78. 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.

79. What is database Trigger?

A database trigger is a PL/SQL block that can defined to automatically execute for insert, update, and delete statements against a table. The trigger can e 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.

80. 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 recovering the database to the time when the export was last performed.

81. 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 recovering the database to the time when the export was last performed.

82. 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.

83. 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

e) Are always in 1NF

84. 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.

85. 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

86. 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

87. 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

88. 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.

89. 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.

90. 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.

91. 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.

92. 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.

93. 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.

94. 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*

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

- 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.

- 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 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.

96. 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 $K = (K - A)^+$

97. 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 Ri is equal to the closure of F. i.e.,

$$((\Pi_{R1}(F)) \cup \ldots \cup (\Pi_{Rn}(F)))^+ = F^+$$

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

98. 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

.**What are the different types of JOIN operations?**

Equi Join: This is the most common type of join which involves only equality comparisons. The disadvantage in this type of join is that there

SQL Questions:

1. Which is the subset of SQL commands used to manipulate Oracle Database structures, including tables?

Data Definition Language (DDL)

2. What operator performs pattern matching?

LIKE operator

3. What operator tests column for the absence of data?

IS NULL operator

4. Which command executes the contents of a specified file?

START <filename> or @<filename>

5. What is the parameter substitution symbol used with INSERT INTO command?

&

6. Which command displays the SQL command in the SQL buffer, and then executes it?

RUN

7. What are the wildcards used for pattern matching?

For single character substitution and % for multi-character substitution

8. State true or false. EXISTS, SOME, ANY are operators in SQL.

True

9. State true or false. !=, <>, ^= all denote the same operation. True

10. What are the privileges that can be granted on a table by a user to others?

Insert, update, delete, select, references, index, execute, alter, all

11. What command is used to get back the privileges offered by the GRANT command?

REVOKE

12. Which system tables contain information on privileges granted and privileges obtained?

USER_TAB_PRIVS_MADE, USER_TAB_PRIVS_RECD

13. Which system table contains information on constraints on all the tables created?

USER_CONSTRAINTS

14. TRUNCATE TABLE EMP;

DELETE FROM EMP;

Will the outputs of the above two commands differ?

Both will result in deleting all the rows in the table EMP.

15. What the difference is between TRUNCATE and DELETE commands?

TRUNCATE is a DDL command whereas DELETE is a DML command. Hence DELETE operation can be rolled back, but TRUNCATE operation cannot be rolled back. WHERE clause can be used with DELETE and not with TRUNCATE.

16. What command is used to create a table by copying the structure of another table?

Answer:

CREATE TABLE AS SELECT command

Explanation:

To copy only the structure, the WHERE clause of the SELECT command should contain a FALSE statement as in the following.

CREATE TABLE NEWTABLE AS SELECT * FROM EXISTINGTABLE WHERE 1=2;

If the WHERE condition is true, then all the rows or rows satisfying the condition will be copied to the new table.

17. What will be the output of the following query?

SELECT REPLACE (TRANSLATE(LTRIM(RTRIM('!! ATHEN !!','!'), '!'), 'AN', '**'),'*','TROUBLE') FROM DUAL;

TROUBLETHETROUBLE

18. What will be the output of the following query?

SELECT DECODE(TRANSLATE('A','1234567890','1111111111'), '1','YES', 'NO');

Answer :

NO

Explanation :

The query checks whether a given string is a numerical digit.

19. What does the following query do?

SELECT SAL + NVL(COMM,0) FROM EMP;

This displays the total salary of all employees. The null values in the commission column will be replaced by 0 and added to salary.

20. Which date function is used to find the difference between two dates?

MONTHS_BETWEEN

21. Why does the following command give a compilation error?

DROP TABLE &TABLE_NAME;

Variable names should start with an alphabet. Here the table name starts with an '&' symbol.

22. What is the advantage of specifying WITH GRANT OPTION in the GRANT command?

The privilege receiver can further grant the privileges he/she has obtained from the owner to any other user.

23. What is the use of the DROP option in the ALTER TABLE command?

It is used to drop constraints specified on the table.

24. What is the value of 'comm' and 'sal' after executing the following query if the initial value of 'sal' is 10000?

*UPDATE EMP SET SAL = SAL + 1000, COMM = SAL*0.1;*

sal = 11000, comm = 1000

25. What is the use of DESC in SQL?

DESC has two purposes. It is used to describe a schema as well as to retrieve rows from table in descending order.

The query SELECT * FROM EMP ORDER BY ENAME DESC will display the output sorted on ENAME in descending order.

26. What is the use of CASCADE CONSTRAINTS?

When this clause is used with the DROP command, a parent table can be dropped even when a child table exists.

27. Which function is used to find the largest integer less than or equal to a specificvalue?

FLOOR

28. What is the output of the following query?

SELECT TRUNC(1234.5678,-2) FROM

DUAL; 1200

SQL HANDSON

Questions Based On Data Definition Language Commands:

1. **Create A Table EMP And DEPT Using The Following Information.**
   a. **DEPT:**

COLUMN NAME	DATATYPE (SIZE)
DEPTNO	NUMBER (2)
DNAME	VARCHAR2 (14)
LOC	VARCHAR2 (14)

   b. **EMP:**

COLUMN NAME	DATATYPE(SIZE)
EMPNO	NUMBER (4)
ENAME	VARCHAR2 (10)
JOB	VARCHAR2 (9)
MGR	NUMBER (4)
HIREDATE	DATE
SAL	NUMBER (7, 2)
COMM	NUMBER (7, 2)
DEPTNO	NUMBER (2)

2. **Check the Default Size of a Number, Char and Date Data types.**
3. **Describe the Structure of the Table and EMP Table.**
4. **Add two columns to the table EMP with the following information in one single**

ALTER COMMAND.

COLUMN NAME	DATATYPE (SIZE)
SEX	CHAR (1)
PLACE	CHAR (15)

5. Modify the column job present in the EMP table with the following information given below:

COLUMN NAME	DATATYPE (SIZE)
JOB	VARCHAR2 (15)

6. modify the column ENAME present in the EMP table with the following information given below:

COLUMN NAME	DATATYPE (SIZE)
ENAME	CHAR (15)

7. Decrease the size for the column EMPNO with the following information:-

COLUMN NAME	DATATYPE (SIZE)
EMPNO	NUMBER (2)

8. Modify the column name of EMPNO to EMPLOYEE_NUMBER present in the EMP table verify the result.

9. Add a new column nationality placed between JOB and MGR columns and verify the result

10. Drop the table DEPT and EMP.

11. What is the data type of the column HIREDATE and how many bytes it occupies.

12. Create a table EMP and DEPT using the following information.

a. DEPT:

COLUMN NAME	DATATYPE(SIZE)	
DEPTNO	NUMBER(2) PRIMARY KEY	CONSTRAINT PK_DEPTNO
DNAME	VARCHAR2(14)	
LOC	VARCHAR2(14)	

b.EMP:

COLUMN NAME	DATATYPE(SIZE)

EMPNO	NUMBER(4)CONSTRAINTPK_EMPNO PRIMARY KEY
ENAME	VARCHAR2(10) CONSTRAINT UQ_DEPTNO UNIQUE
JOB	VARCHAR2(9)
MGR	NUMBER(4)
HIREDATE	DATE DEFAULT SYSDATE
SAL	NUMBER(7,2) CONSTRAINT CK_SAL CHECK(SAL)
COMM	NUMBER(7,2)
DEPTNO	NUMBER(2)CONSTRAINTFK_DEPTNO REFERENCE

13. Select all the constraints in the EMP table

SOL: SELECT * FROM USER_CONSTRAINTS WHERE TABLE_NAME='EMP';

14. select the owner, constraints name, constraints type, table name, status for DEPT table

SOL: SELECT OWNER, CONSTRAINTS_ NAME, CONSTRAINTS_TYPE, TABLE_ NAME, STATUS FROM USER_CONSTRAINTS WHERE TABLE_NAME='DEPT';

15. Drop the constraints UQ_FMANE from EMP table

SOL: ALTER TABLE EMP DROP CONSTRAINT UQ_FNAME;

16. Add a new column PINCODE with not null constraints to the existing table DEPT

SOL: ALTER TABLE DEPT ADD(PINCODE NUMBER(6) NOT NULL);

17. Disable the constraints PK_DEPTNO present in the DEPT table

SOL: ALTER TABLE DEPT DISABLE CONSTRAINTS PK_DEPTNO;

18. Enable the constraints PK_DEPTNO which is defined in the DEPTNO column of DEPT table;

SOL: ALTER TABLE DEPT ENABLE CONSTRAINTS PK_DEPTNO;

19. Insert the given values into the tables:

EMP:

(i) 7369, SMITH, CLERK, 7902, 17 – DEC – 80, 800, NULL, 20

(ii) 7499, ALLEN SALEMAN, 7698, 20 – FEB – 81, 1600, 300, 30

(iii) 7521, WARD, SALESMAN, 7698, 22 – FEB – 81, 1250, 500, 30

(iv) 7566, JONES, MANAGER, 7839, 02 – APR – 81, 2975, NULL, 20

(v) 7654, MARTIN, SALESMAN, 7698, 28 – SEP – 81, 1250, 1400, 30

(vi) 7698, BLAKE, MANAGER, 7839, 01 – MAY – 81, 2850, NULL, 30

(vii) 7782, CLERK, MANAGER, 7839, 09 – JUN – 81, 2450, NULL, 10

(viii) 7788, SCOTT, ANALYST, 7566, 19 – NOV – 96, 3000, NULL, 20

(ix) 7839, KING, PRESIDENT, NULL, 17 – NOV – 81, 5000, NULL, 10

(x) 7844, TURNER, SALESMAN, 7698, 08 –SEP – 81, 1500, 0, 30

(xi) 7876, ADAMS, CLERK , 7788, 23 – DEC – 96, 1100, NULL, 20

(xii) 7900, JAMES, CLERK, 7698, 03 – DEC – 81, 950, NULL, 30

(xiii) 7902, FORD, ANALYST, 7566, 03 – DEC – 81, 3000, NULL, 20

(xiv) 7934, MILLER, CLERK, 7782, 23 – JAN – 82, 1300, NULL, 10

(xv) 7943, JOHN, CLERK, 7943, 10 – DEC – 83, 2000, NULL, 50

DEPT:

(i) 10, ACCOUNTING, NEW YORK

(ii) 20, RESEARCH, DALLAS

(iii)30, SALES, CHICAGO

(iv)40, OPERATIONS, BOSTON

(v) 50, COMPUTER, AMERICA

20. Insert only the records of employee number, name, salary into EMP table

SOL: INSERT INTO EMP (EMPNO, ENAME, SAL) VALUES (_7955‘, _PAUL‘, 1200);

21. insert two rows into EMP table using parameter substitution

SOL: INSERT INTO EMP VALUES (&EMPNO, _&ENAME‘, _&JOB‘, &MGR, _&IIIREDATE‘, &SAL, &COMM, &DEPTNO);

22. insert the current transaction date temporary table

SOL: INSERT INTO TEMP VALUES (SYSDATE);

Problems on Select Command:

23. List the Information of all Employees

SOL: SELECT * FROM EMP;

24. List the information of all the departments

SOL: SELECT * FROM DEPT;

25. LIST THE DEPARTMENT NUMBERS, EMPLOYEE NUMBERS AND THEIR MANAGERS NUMBERS

SOL: SELECT DEPT. DEPTNO, EMP. EMPNO, EMP. MGR FROM EMP, DEPT;

26. List department name and locations from DEPT table

SOL: SELECT DNAME, LOC, FROM DEPT;

27. List the information of employees and their departments in a single DMI command SOL: SELECT *. FROM EMP;

28. Copy all the records of their columns EMPNO, ENAME, JOB from EMP table and insert the records into a temp table with column names same as EMPNO, ENAME, JOB

SOL: INSERT INTO TEMP (EMPNO, ENAME, JOB) SELECT EMPNO, NAME, JOB

29. List the details of both the tables

SOL: SELECT * FROM EMP, DEPT;

30. List the information of all the employees present in the user named SCOTT

SOL: SELECT * FROM SCOTT. EMP;

31. List the information of the departments from your BATCHMATES DEPT table

SOL: SELECT * FROM ORA252P. DEPT;

32. List out the table names in your schema

SOL: SELECT * FROM TAB;

33. List all the system tables

SOL: SELECT *FROM SYS. DICTIONARY;

34. Get the information of the maximum available blocks allotted to a particular user for creating tables from the system tables

SOL: SELECT * FROM USER_TS_QUOTAS;

35. List out all the privileges given to a particular user

36. List out all the tables which start with 'S'

SOL: SELECT TABLE_NAME FROM USER_TABLES WHERE TABLE_NAME LIKE _S%‘;

37. Copy the structure of dept table alone to a temporary table called TEMP1

SOL: CREAT TABLE TEMP1, AS SELECT * FROM DEPT WHERE 1=2;

Problems on update command:

38. Update the salary by 10% hike to analysts working in department number 20 and 30

SOL: UPDATE EMP SET SAL = SAL + 0. 1 WHERE DEPTNO IN (10,20) AND JOB =_ANALYST‘;

39. Give 5% raise in salary to all the salesman

SOL: UPDATE EMP SET SAL=SAL+.5 WHERE JOB = _SALESMAN‘;

40. PROMOTE ALL THE EMPLOYEES DISCRIBED AS SALESMAN TO SALES OFFICER IF THEIR GROSS SALARY PER MOUNTH IS GREATER THEN 3000 SOL:

UPDATE EMP SET JOB = _SALESOFFICER‘ WHERE JOB = _SALESMAN‘ AND SAL>3000;

41. Give all the employees of commission of rs500

SOL: UPDATE EMP SET COMM = 500;

42. Change the department of JAMES to 20

SOL: UPDATE EMP SET DEPTNO = 20 WHERE ENAME = _JAMES‘;

43. Calculate all the employees total salary with commission

SOL: SELECT SALES SAL +NVL (COMM) ―TOTAL‖ FROM EMP;

Problems on delete command:

44. Delete all the records of employees

SOL: DELETE FROM EMP;

45. Get back the original records of employees back

SOL: ROLLBACK;

46. Allen’s record only

SOL: DELETE FROM EMP WHERE ENAME = _ALLEN‘; **47. Delete records of ename column only and verify it**

SOL: NOT POOSIBLE

48. Delete the records of employee number 7782

SOL: DELETE FROM EMP WHERE EMPNO=7782;

49. Delete the employee's records who doesn't have commission

SOL: DELETE FROM EMP WHERE COMM IS NULL;

50. Get back the original records back

SOL: ROLLBACK;

51. Delete the duplicate records of the employee table

SOL: DELETE FROM EMP A WHERE ROWID<> (SELECT MIN (ROWID) FROM EMP B WHERE A. EMPNO=B. EMPNO);

52. Delete the first five records of employee table

SOL: DELETE FROM EMP X WHERE 5 > (SELECT COUNT (ROWID) FROM EMP Y WHERE Y. ROWID < X.ROWID);

53. Delete the rows of the temp table permanently

SOL: TURNCATE TABLE TEMP;

Problems on transactional commands:

54. Update a record of EMP table and save the changes permanently in the database

SOL: UPDATE EMP SET SAL=SAL+100 WHERE EMPNO=100; COMMIT;

55. Sql * plus has the facility to automatically save all the records without issuing the TCL command which is that?

SOL: SET AUTOCOMMIT ON

56. Give all the privileges you have of a database object to another

SOL: GRANT ALL ON EMP TO ORA253A;

57. Give only select, insert privileges to another user

SOL: GRANT SELECT, INSERT ON EMP TO ORA267A;

58. List the user's id and which database object you have granted

SOL: SELECT * FROM USE_TAB_PRIVS;