

BABASAHEB BHIMRAO AMBEDKAR BIHAR UNIVERSIY

ASSIGNMENT REPORT

BCA – 305

LAB ON DBMS (SQL/MS-ACCESS)

Name	: - Saurabh Kumar
Class	: - BCA 3 rd Semester
Roll No.	: - 2010068
Registration No.	: - 19BAVGCS061/19
Session	: - 2019 – 2022
College	: - L.S. College, Muzaffarpur
Topic	: - 10 Questions Based on SQL

INDEX

PAGE NO.:	<input checked="" type="checkbox"/>

S.No.	TOPIC	Page No.
01.	What is SQL & Important Features of SQL?	01-04
02.	Write SQL command to Create table:- Student (Std-roll, Std-name, Std-add)	05
03.	Write SQL command to Insert five records in Student table.	06
04.	What is ALTER command? Write the Syntax & Example of ALTER Command.	07-08
05.	What is Join and different types of Join?	09-13
06.	Explain Five aggregate function in SQL	14-17
07.	What is views & types of views?	18-20
08.	Write the differences between Equi Join and Natural Join.	Q1-Q2
09.	What is Sub-queries and types of Sub-queries?	Q3-Q5
10.	What is Indexes and Synonyms in SQL	Q6-Q7



Q1. What is SQL and important features of SQL?

Ans: The SQL Stands for Structured Query Language. It is a type of computer language for storing, manipulating and retrieving data stored in relational database. The SQL allows a user to communicate with Relational database and retrieve data from their database or table and manipulate the table or stored data in the table. The SQL is introduced by E.F. Codd in 1970 and it is firstly implemented in Oracle in 1979. The SQL is mainly used to perform different operations on the data stored in table of database, such as creating database or table, inserting records in the table, updating records, creating views and retrieving different information from the data stored in the table. The Structured Query language is the standard language for relational database management system like MySQL, Oracle, MS-Access SQL Server, DB2 etc. The SQL is originally called SEQUEL (Structured English Query Language).

The Structured Query language is accepted by American National Standard Institute (ANSI) in 1986 and International organization for Standardization (ISO) in 1987.

The Structured Query language is based on the Relational Algebra and Tuple Relational Calculus.

Using SQL Commands on the database or table we can perform different operations in easy and fast way which take many more time to do manually. The Structured

Query Language is also known or categorized as non-procedural language because SQL allows to declare what to do but not how to do that tasks. It is a domain-specific language used

in programming and designing and manipulating data. The SQL is also categorized into 4 different Sub-SQL languages

These are DDL (Data Definition Language), DML (Data manipulation Language), DCL (Data Control Language), TCL (Transaction Control language).

⇒ Important Features of SQL.

The Structured Query Language is full of its important and useful features. Some of the useful & important features of SQL are as:-

- (i) The Standard database applications use SQL for execute queries against a database.
- (ii) SQL can create new database or new table.
- (iii) It can store, update, delete data from the table.
- (iv) It can create different views of the records according to the user need.
- (v) It helps user to modify database, table or columns in the table.
- (vi) The SQL language is easy to learn and implement or use.
- (vii) The SQL language provide different features to store data safely over a long period of time.

- (viii) The SQL language does not require a large number of coding lines for managing the database system and records in the database.
- (ix) The SQL queries can be used to retrieve large amounts of records or information from the records quickly & efficiently.
- (x) The SQL query language can be easily used in desktop, tablet, laptop and even in smartphone.
- (xi) The SQL language supports multi-user environment so, that multiple user can use same table or database at same time.
- (xii) The SQL provides a uniform platform worldwide to all its users.
- (xiii) The SQL command follows only limited rules which is easy to remember.
- (xiv) The SQL provides different aggregate functions to do different operation over records.

X — X — X — X — X — X — X — X — X

Q2. Write SQL Command to Create Following Table:-

Student (Std-roll, Std-name, Std-add).

Ans:- The SQL command for creating Student table are as:-

```
mySQL > CREATE TABLE STUDENT
      → (
      → Std-roll int(5) PRIMARY KEY,
      → Std-name varchar(20) NOT NULL,
      → Std-addl varchar(30)
      → );
```

OR -> Query ok, 0 row affected (0.51 sec)

Effect ->

STUDENT		
Std-roll	Std-name	Std-addl
X — X — X — X — X — X — X — X — X		

Q3. Write SQL Command to Insert five records in Student table

Ans:- The SQL command to Insert five record in Student table are :-

mySQL > INSERT INTO STUDENT (Std-roll,
Std-name, Std-add)

→ VALUES (02, 'Saurabh', 'muzaffarpur'),
(06, 'Rakesh', 'Patna'), (08, 'Rita',
'Varanasi'), (07, 'muresh', 'Pune'),
(10, 'Vaishnavi', 'Tirumantpuram');

QIP → Query OK, 5 row affected (0.32 sec)

Effect:->

STUDENT		
Std-roll	Std-name	Std-add
02	Saurabh	Muzaffarpur
06	Rakesh	Patna
08	Rita	Varanashi
07	muresh	Pune
10	vaishnavi	Tirumantpuram

X — X — X — * — X — X — X — X

Q4 What is ALTER Command? Write the Syntax and example of ALTER command.

Ans:- The ALTER command is a type of SQL Data Definition Command (DDL). It is used to modify the structure of the existing table. Using ALTER Command we can add, delete, modify or rename table or columns in the existing table.

Syntax :-
example for
adding new
column

ALTER TABLE tablename
ADD Columnname datatype(size)
constraint;

Example:-

ALTER TABLE Student
ADD Course VARCHAR(10) NOTNULL;

Syntax & example for modify column -

Syntax:-

ALTER TABLE Tablename
MODIFY Columnname datatype(size)
constraint;

Example:-

```
ALTER TABLE Student  
MODIFY course varchar(10);
```

Syntax & example for Rename Columnname.

Syntax:-

```
ALTER TABLE tablename  
RENAME column OldColumnName  
TO NewColumnName;
```

Example:-

```
ALTER TABLE Student  
RENAME Column Course TO Subject;
```

Syntax & example for Delete or Remove existing Column.

Syntax:-

```
ALTER TABLE tablename  
DROP Columnname;
```

Example:-

```
ALTER TABLE Student  
DROP Subject;
```

X—X—X—X—X—X—X—X—X—X—X

Q5. What is Join and different types of Join.

Ans.: The Join is used to combine records from two or more tables in a single database. A join is a means for combining field from two table by using values common to each. The join pair two tuples from different table if and only if a given join condition satisfied. In Relation algebra it is represented by (\bowtie).

There are 2 types of join:-

- (i) INNER Join
- (ii) OUTER Join.

(i) INNER Join \rightarrow An INNER Join include only those records from the both table that satisfy the matching criteria, while the rest of records from the both table are eliminated.

There are 3 types of inner join

- (a) Theta Join
- (b) Equi Join
- (c) Natural Join.

(a) Theta Join \rightarrow It Combines records from the different table of same database which satisfy the given condition. It uses

different types of operator such as
 $<$, $>$, \leq , \geq , $=$, \neq etc.

Syntax & Example of Theta Join.

Syntax:-

```
SELECT columnname FROM tablename1
INNER JOIN tablename2 ON
tablename1.column (condition)
tablename2.column;
```

Example:-

```
SELECT * FROM Student
INNER JOIN marks ON
Student.rollno = marks.rollno;
```

(b) EQUI JOIN:- The Equi Join is a type of Inner Join which uses only equivalence ($=$) condition. It join two tables on the basis of equality of values of common attribute (column). The Equi Join show the matching result column two times one from the 1st table and one from the 2nd table.

Syntax & Example

Syntax:-

```
SELECT columnname FROM table1 INNER
JOIN table2 ON table1.column =
table2.column;
```

Example:-

```
SELECT INFO.Rolno, MARKS.Sem1
FROM INFO INNER JOIN MARKS
ON INFO.Rolno = MARKS.Rolno;
```

(c) Natural Join :- The Natural Join is a type of INNER Join which do not use any operator to join two table. They join two table on the basis of same column name their attribute and domain. The Natural Join do same as Equi Join but it does not show the matching column two times.

Syntax & Example

Syntax:-

```
SELECT Columnname FROM table1
NATURAL JOIN table2;
```

Example:-

```
SELECT * FROM INFO NATURAL JOIN
MARKS;
```

(d) Outer Join :- The Outer Join is a type of Join which include all the tuples (records) from the both table either it is matched or unmatched. The Outer Join are classified into 3 types:-

- (a) Left outer join
- (b) Right outer join
- (c) Full outer join.

(a) LEFT OUTER JOIN:- The LEFT OUTER JOIN is a type of outer join which return all the records (rows) from the LEFT table, with matching result from the right table. The rows which do not match from the right table return the null.

Syntax & Example:-

Syntax:-

```
SELECT columnname FROM left
tablename LEFT JOIN right
tablename ON columnname = columnname;
```

Example:-

```
SELECT * FROM INFO LEFT JOIN
MARSIS ON INFO.xname = MARSIS.
xname;
```

(b) Right outer join:- The Right outer join is a type of outer join which return all the rows from the right table along with matching result from the left table. The rows which do not match the rows from left table

return NULL;

Syntax & Example:-

Syntax:-

```
SELECT columnnames FROM Left Table Name
RIGHT JOIN Right Table name ON
columnname = columnname;
```

Ex:- SELECT * FROM INFO RIGHT JOIN
marks ON INFO.Roll = marks.Roll;

(c) Full Outer Join :- The Full Outer Join is the combination of Left Outer Join & Right Outer Join. The Full Outer Join return all the rows from the both table and it return NULL for the unmatched records.

Syntax & Example:-

Syntax:-

```
SELECT columnnames FROM table1 FULL
JOIN table2 ON columnname1 =
columnname2;
```

Example:-

```
SELECT * FROM INFO FULL JOIN Marks
ON INFO.Rollno = Marks.Rollno;
```



Q6. Explain five aggregate functions in SQL.

Ans.: The aggregate functions perform calculation on the set of values, records & return a single value most of the aggregate function ignore null values except count()

The most common 3 aggregate fun. in SQL are:-

- (i) COUNT()
- (ii) SUM()
- (iii) AVG()
- (iv) MAX()
- (v) MIN()

(i) COUNT() → The aggregate function count return the total no. of rows or records in the table. In this function we can also include the specific condition to count the records.

Syntax & Example:

Syntax:-

```
SELECT COUNT (columnname) FROM
Table name WHERE Condition;
```

Example:-

- (i) SELECT COUNT(*) FROM INFO;
- (ii) SELECT COUNT (rows) FROM INFO WHERE Rows > 100;
- (iii) SUM() → The aggregate function sum() return the total sum of selected numeric values from rows. The sum function also uses the condition for specific purpose.

Syntax & Example:-

Syntax:-

SELECT Sum(columnname) FROM tablename
WHERE Condition;

Example:-

- (i) SELECT Sum(Fee) FROM Student;
- (ii) SELECT Sum(Fee) FROM Student
WHERE Section = 'A';
- (iii) SELECT Sum(Fee) FROM Student
WHERE Fee > 10000;

(iii) $\text{AVG}()$ → The aggregate function $\text{SUM}()$ return the average of selected column values. basically it sum all the values from the selected column and then divide by the total no. of values. we can also include the specific condition in $\text{AVG}()$ function.

Syntax & Example:-

Syntax:-

```
SELECT AVG(columnname) FROM
tablename WHERE Condition;
```

Example:-

(i) `SELECT AVG(feess) FROM INFO;`

(ii) `SELECT AVG(feess) FROM INFO WHERE
Section = 'A';`

(iv) $\text{MAX}()$ → The aggregate function $\text{MAX}()$ return the maximum value from the selected records. we can also include the specific condition in the $\text{MAX}()$ aggregate function.

Syntax & Example:-

Syntax:- `SELECT max(columnname)
FROM tablename WHERE
Condition;`

Example :-

- (i) `SELECT MAX(FEES) FROM INFO;`
- (ii) `SELECT MAX(FEES) FROM INFO
WHERE Rollno > 100;`

(v) `MIN()` → The aggregate function `MIN()` return minimum value from the selected records. We can also include the specific condition for selecting the minimum value from the records. It is the opposite of the `MAX()` aggregate function.

Syntax & Example :-

Syntax :-

```
SELECT MIN(columnname) FROM tablename  
WHERE Condition;
```

Example :-

- (i) `SELECT MIN(FEES) FROM INFO;`
- (ii) `SELECT MIN(FEES) FROM INFO
WHERE Rollno < 100;`

X — X — X — X — X — X — X — X

Q7. What is views & types of views ?

Ans In some cases it does not desirable for all the user to see the entire logical model or entire table or records stored in the table. In this condition the SQL provides a specific feature known as 'views'. The views are the virtual relation or virtual table through which selective portion of the data one or more table can be seen or selected. Views do not contain their own data and it also does not exist physically in the database. It is the only logical or virtual concept which is very useful for the user to do more task in less time. If any modification like insertion, delete, update are done on the views it also affect the actual table of the database on which the view is based. The views which is defined by the user for their specific purpose are stored in the data dictionary in the table called USER_VIEWS. The view is also treated as a table. All the operation which we can do upon the table can also be done on the views.

CREATING view

Syntax & Example :-

Syntax:-

```
CREATE VIEW viewname AS
SELECT Columnname --- FROM
tablename WHERE CONDITION;
```

Example:-

- (i) CREATE VIEW view1 AS
SELECT name, roll, mobile FROM INFO;
- (ii) CREATE VIEW view2 AS
SELECT * FROM INFO
WHERE Rooms > 20;
- (iii) CREATE VIEW BCA AS
SELECT * FROM INFO
WHERE COURSE = 'BCA';

MODIFICATION IN View

Syntax:- ALTER VIEW Viewname AS
SELECT Columnname FROM tablename
WHERE Condition

Example:- ALTER VIEW BCA AS SELECT
name, Rollno FROM INFO;

Operations on Views

Syntax & Example

Syntax :-

UPDATE viewname

SET columnname = value ,

columnname = value , . . .

WHERE condition;

Example :-

(i) UPDATE view

SET Rollno = Rollno + 10

WHERE course = 'BCA';

(ii) UPDATE view 1

SET Rollno = Rollno - 10 ;

DELETE View

Syntax & Example

Syntax :-

DROP view viewname;

Example :-

DROP view view1;



Q8: Write the differences between Equi Join & natural Join.

Ans: The differences between Equi Join & natural Join are as:-

- (i) Natural Join Joins two tables based on same attribute name & datatype
Equi Join Joins two tables on the basis of column which is explicitly specified in the ON clause.
- (ii) In the natural Join the resulting table will contain all the attributes of both the table but keep only one copy of each common column.
In the Equi Join the resulting table will contain all the attribute of both table including common column also.
- (iii) Natural Join do not use any operator to join two tables.
Equi Join use "=" equality operator to join two tables.
- (iv) In natural Join we do not specify any column on which we want to join
In Equi Join we specify the column name to join two tables.

(iv) Syntax:-

Natural Join:-

```
SELECT Columnnames FROM table1
NATURAL JOIN table2;
```

Equi Join:-

```
SELECT Columnnames FROM table1
INNER JOIN table2 ON
table1.column = table2.column;
```

(v) Example:-

Natural Join:-

```
SELECT * FROM INFO NATURAL
JOIN MARKS;
```

Equi Join:-

```
SELECT * FROM INFO INNER JOIN
MARKS ON INFO.roll = marks.roll;
```

(vi) O/P →

Natural Join

roll	Name	marks
25	Rahul	62
36	Ravesh	54
79	Salmi	29
02	Arum	74

Equi Join

roll	name	roll	marks
25	Rahul	25	62
36	Ravesh	36	54
79	Salmi	79	29
02	Arum	02	74

X — X — X — X — X — X — X — X

Q9. What is Sub-queries & types of Sub-queries?

Ans:- A Sub-queries is defined as a query within a query. Sub-queries enables you to write queries that select data rows or criteria that are actually developed while the query is executed at run time. A Sub-queries is used in any place where an expression is allowed providing it returns a single value can also be listed as an object in a From clause listing. The Sub-queries can be nested inside a SELECT, INSERT, DELETE, UPDATE or inside another subqueries. A Subqueries is usually added within the WHERE clause of another SQL Select Statement. The Inner Sub query executed first before its parent query so that the result of an inner query can be passed to the outer query. A subquery is also called an Inner query or Inner Select while the statement containing a subquery called an outer select or outer query.

The Sub queries are mainly classified into 3 types:-

- (i) Single Row Sub query
- (ii) Multiple Row Sub query
- (iii) Correlated Sub query.

(i) Single Row Subquery:-

⇒ A Single Row Subquery is a subquery which returns single row output. They make the use of single row comparison operators, when used in WHERE clause.

A Single Row Subquery is used when the outer query are based on a single unknown value of the inner query.

Example:-

```
SELECT Fname, Salary, Dept
FROM emp
WHERE Salary = (SELECT MIN(Salary)
FROM emp);
```

(ii) Multiple Row Subquery

⇒ A multiple Row Subquery is a subquery which returns multiple row output.

The make use of multiple row comparison operators like IN, ANY, ALL etc. The multiple Row Subquery returns more than one row of results to the parent query. Multiple Row Subquery are used most commonly in WHERE CLAUSE and HAVING CLAUSE. There can be the queries returning multiple column also treated as the multiple row subqueries.

Example :-

```
SELECT name, deptid, empid FROM emp
WHERE deptid IN (SELECT deptid FROM
dept WHERE locid = 100)
```

(iii) Correlated Sub Query :- The correlated Sub query is a type of Sub query which depends upon the data provided by the Outer Query or Parent Query. This type of Sub query also includes that use the EXISTS operator to test the existence of data rows satisfying specific criteria. A correlated Sub query can produce result tables that answer complex management questions. In a correlated Sub query the inner query is executed repeatedly, once for each row that might be selected by the outer query.

Example :-

```
SELECT * FROM emp1
WHERE salary > (SELECT avg(salary) FROM
emp2 WHERE
emp1.deptid = emp2.dept)
```

X—X—X—X—X—X—X—X—X

10. What is Indexes & Synonyms in SQL?

Ans:- INDEXES

⇒ Indexes are the database objects that are used to tune the performance of SELECT Query. An index is created on table column, which store all the values of the column under index segment. In simple word we can say that indexes are special lookup tables that the database search engine can use to speed up data retrieval. An Index is a pointer to data in the table. An index in a database is very similar to an index in the book.

An index helps to speed up the SELECT & WHERE Statement but it slow down the data input with INSERT and UPDATE Statement.

CREATE INDEX:-

⇒ CREATE INDEX indexname ON tablename;
ex:- CREATE INDEX S_no. ON emp;

DELETE INDEX:-

⇒ DROP INDEX indexname;
ex:- DROP INDEX S_no;

Synonyms:-

- A Synonyms is an alias, that is a form of shorthand used to simplify the task of referencing a database. The Synonyms is a redefinable label for a table which we can use as an alternative to the actual table name.
- A Synonyms provide another name for database objects, referred to as original object, that may exist on a local or another server. The name of every Synonyms should be unique.
- The Synonyms are database dependent means for every database there Synonyms exist and it cannot be accessed by other database.

CREATE Synonyms:-

CREATE PUBLIC Synonym Synonymname
FOR Object

ex:- CREATE PUBLIC Synonym Emp FOR
Employee.empid;

DELETE Synonyms:-

DROP Synonyms Emp;

