Downloaded from www.studiestoday.com

Unit-III
DATABASES MANAGEMENT SYSTEM AND SQL

DBMS & Structured Query Language Chapter: 07

➢ Basic Database concepts

Data : Raw facts and figures which are useful to an organization. We cannot take decisions on the basis of data.

Information: Well processed data is called information. We can take decisions on the basis of information

Field: Set of characters that represents specific data element.

Record: Collection of fields is called a record. A record can have fields of different data types.

File: Collection of similar types of records is called a file.

Table: Collection of rows and columns that contains useful data/information is called a table. A table generally refers to the passive entity which is kept in secondary storage device.

Database: Collection of logically related data along with its description is termed as database.

Relation: Relation (collection of rows and columns) generally refers to an active entity on which we can perform various operations. It is also known as table.

Tuple: A row in a relation is called a tuple. It is also known as record.

Attribute:A column in a relation is called an attribute. It is also termed as field or data item.

Degree: Number of attributes in a relation is called degree of a relation.

Cardinality: Number of tuples in a relation is called cardinality of a relation.

Primary Key: Primary key is a key that can uniquely identifies the records/tuples in a relation. This key can never be duplicated and NULL.

Foreign Key: Non key attribute of a table acting as primary key in some other table is known as Foreign Key in its current table. This key is used to enforce referential integrity in RDBMS.

Candidate Key: Attributes of a table which can serve as a primary key are called candidate keys.

Alternate Key: All the candidate keys other than the primary key of a relation are alternate keys for a relation.

DBA: Data Base Administrator is a person (manager) that is responsible for defining the data base schema, setting security features in database, ensuring proper functioning of the data bases etc.

Select Operation: The select operation selects tuples from a relation which satisfy a given condition. It is denoted by lowercase Greek Letter σ (sigma).

Project Operation: The project operation selects columns from a relation which satisfy a given condition. It is denoted by lowercase Greek Letter π (pi). It can be thought of as picking a sub set of all available columns.

Union Operation: The union (denoted as ∪) of a collection of relations is the set of all distinct tuples in the collection. It is a binary operation that needs two relations.

Set Difference Operation: This is denoted by – (minus) and is a binary operation. It results in a set of tuples that are in one relation but not in another

➢ Structured Query Language

SQL is a non procedural language that is used to create, manipulate and process the databases(relations).

Downloaded from www.studiestoday.com

Downloaded from www.studiestoday.com

1. Data Definition Language (DDL)

DDL contains commands that are used to create, modify or remove the tables, databases, indexes, views, sequences and synonyms etc.

e.g: Create table, create view, create index, alter table,drop table etc.

2. Data Manipulation Language (DML)

DML contains commands that can be used to manipulate the data base objects and to query the databases for information retrieval.

e.g Select, Insert, Delete, Update etc.

3. **Transaction Control Language (TCL)**

TCL include commands to control the transactions in a data base system. The commonly used commands in TCL are COMMIT, ROLLBACK etc.

➢ **Operators in SQL:**The following are the commonly used operators in SQL

1.	Arithmetic Operators	+, -, *, /
2.	Relational Operators	=, <, >, <=, >=, <>
3.	Logical Operators	OR, AND, NOT

➢ **Data types of SQL:** Just like any other programming language, the facility of defining data of various types is available in SQL also. Following are the most common data types of SQL.

1)	NUMBER	e.g. Number(n,d) Number (5,2)
2)	CHAR	CAHR(SIZE)
3)	VARCHAR / VARCHAR2	VARCHAR2(SIZE)
4)	DATE	DD-MON-YYYY

➢ **Constraints:**Constraints are the conditions that can be enforced on the attributes of a relation. The constraints come in play whenever we are trying to insert, delete or update a record in a relation.

Not null ensures that we cannot leave a column as null. That is a value has to be supplied for that column.

e.g. name varchar(25) not null

Unique constraint means that the values under that column are always unique.

e.g. Roll_no number(3) unique

Primary key constraint means that a column cannot have duplicate values and not even a null value.

e.g. Roll_no number(3) primary key

The main difference between unique and primary key constraint is that a column specified as unique may have null value but primary key constraint does not allow null values in the column.

Foreign key is used to enforce referential integrity and is declared as a primary key in some other table.

e.g. cust_id varchar(5) references master(cust_id)

it declares cust_id column as a foreign key that refers to ***cust_id*** field of table ***master***.

That means we cannot insert that value in cust_id filed whose corresponding value is not present in cust_id field of master table. Moreover we can't delete any row in ***master*** *table , if a corresponding value of cust_id field is existing in the dependent table.*

Check constraint limits the values that can be inserted into a column of a table.

Downloaded from www.studiestoday.com

Downloaded from www.studiestoday.com

e.g. marks number(3) check(marks>=0)

The above statement declares marks to be of type number and while inserting or updating the value in marks it is ensured that its value is always greater than or equal to zero.

Default constraint is used to specify a default value to a column of a table automatically. This default value will be used when user does not enter any value for that column.

e.g. balance number(5) default = 0

➢ SQL COMMANDS :

1. **Create Table command** is used to create a table . The syntax of this Command is:

```
CREATE TABLE <Table_name>
     ( column_name 1    data_type1 [(size)  column_constraints],
       column_name 1    data_type1 [(size)  column_constraints],
        :
        :
        [<table_constraint> (column_names)] );
```

2. **The ALTER Table command**is used to change the definition (structure) of existing table.

```
ALTER TABLE <Table_name>ADD/MODIFY <Column_defnition>;    For Add or modify column
ALTER TABLE <Table_name> DROP COLUMN <Column_name>;       For Deleting a column
```

3. **The INSERT Command:**The rows (tuples) are added to a table by using INSERT command. The syntax of Insert command is:

```
INSERT INTO <table_name> [(<column_list>)]VALUES (<value_list>);
```

e.g.,

```
INSERT INTO EMP (empno, ename, sex, sal, deptno) VALUES(1001, 'Ravi', 'M', 4500.00, 10);
```

If the order of values matches the actual order of columns in table then it is not required to give the column_list in INSERT command. e.g.

```
INSERT INTO EMP  VALUES(1001, 'Ravi', 'M', 4500.00, 10);
```

4. **The Update command** is used to change the value in a table. The syntax of this command is:

```
UPDATE <table_name>
SET column_name1=newvalue1/expression [,column_name2=newvalue2/expression,……]
WHERE <condition>;
```

e.g., to increase the salary of all the employees of department No 10 by 10% , then command will be:

```
UPDATE emp
SET sal=sal*1.1
WHERE Deptno=10;
```

5. **The DELETE command** removes rows from a table. This removes the entire rows, not individual field values. The syntax of this command is

```
DELETE FROM <table_name>
[WHERE <condition>];
```

e.g., to delete the tuples from EMP that have salary less than 2000, the following command is used:

```
DELETE FROM emp WHERE sal<2000;
```

To delete all tuples from emp table:

```
DELETE FROM emp;
```

6. **The SELECT command** is used to make queries on database. A query is a command that is given to produce certain specified information from the database table(s). The SELECT command

Downloaded from www.studiestoday.com

Downloaded from www.studiestoday.com

can be used to retrieve a subset of rows or columns from one or more tables. The syntax of Select Command is:

```
SELECT <Column-list>
FROM <table_name>
[WHERE<condition>]
[GROUP BY <column_list>]
[HAVING<condition>]
[ORDER BY <column_list [ASC|DESC ]>]
```

The **select** clause list the attributes desired in the result of a query.
e.g.,To display the names of all Employees in the *emp* relation:

select *ename* **from** *emp;*

To force the elimination of duplicates, insert the keyword **distinct** after select**.**
Find the number of all departments in the *emp* relations, and remove duplicates

select **distinct***deptno* from *emp;*

An asterisk (*) in the select clause denotes "all attributes"

SELECT* **FROM** *emp;*

The **select** clause can contain arithmetic expressions involving the operation, +, –, *, and /, and operating on constants or attributes of tuples.The query:

SELECT*empno, ename, sal* * 12 **FROM** ***emp;***

would display all values same as in the *emp* relation, except that the value of the attribute *sal* is multiplied by 12.
The WHERE clause in SELECT statement specifies the criteria for selection of rows to be returned.

- **Conditions based on a range (BETWEEN Operator):**The Between operator defines a range of values that the column values must fall in to make condition true . The range includes both lower value and upper value.

e.g., Find the empno of those employees whose salary between 90,000 and 100,000 (that is, 90,000 and 100,000)

SELECT *empno* **FROM** *emp* **WHERE** *sal***BETWEEN** 90000 **AND** 100000;

- **Conditions based on a list (IN operator):** To specify a list of values , IN operator is used. IN operator selects values that match any value in a given list of values.

For example , to display a list of members from 'DELHI', 'MUMBAI', 'CHENNAI' or'BANGALORE' cities :
SELECT * FROM members WHERE city IN ('DELHI', 'MUMBAI', 'CHENNAI' ,'BANGALORE') ;

The **NOT IN** operator finds rows that do not match in the list. So if you write
SELECT * FROM members WHERE city NOT IN ('DELHI', 'MUMBAI', 'CHENNAI' , 'BANGALORE') ;

It will list members not from the cities mentioned in the list.

- **Conditions based on Pattern:** SQL also includes a string-matching operator, LIKE, for comparison on character string using patterns. Patterns are described using two special wildcard characters:
    - Percent (%) – '%' matches any substring(one,more than one or no character).
    - Underscore (_) – '_' character matches exactly one character.
  - Patterns are case-senstive.
  - **Like** keyword is used to select row contaning columns that match a wildcard pattern.
  - The keyword **not like** is used to select the row that do not match the specified patterns of characters.
- **Searching for NULL:**The NULL value in a column is searched for in a table using *IS NULL* in the WHERE clause(Relational Operators like =,<> etc can not be used with NULL).

Downloaded from www.studiestoday.com

Downloaded from www.studiestoday.com

For example, to list details of all employees whose departments contain NULL (i.e., novalue), you use the command:

SELECT empno, ename FROM emp Where Deptno **IS NULL**;

- **ORDER BY Clause:** Whenever a select query is executed the resulting rows are displayed in the order in which the exist in the table. You can sort the result of a query in a specific order using ORDER BY clause. The ORDER BY clause allow sorting of query result by one or more columns. The sorting can be done either in ascending or descending order.

Note:- If order is not specifies that by default the sorting will be performed in ascending order.

- **GROUP BY Clause:** The GROUP BY clause groups the rows in the result by columns that have the same values.Grouping is done on column name. It can also be performed using aggregate functions in which case the aggregate function produces single value for each group.
- **Aggregate Functions:** These functions operate on the multiset of values of a column of a relation, and return a value

avg:	average value
min:	minimum value
max:	maximum value
sum:	sum of values
count:	number of values

These functions are called aggregate functions because they operate on aggregates of tuples. The result of an aggregate function is a single value.

- **HAVING Clause:** The HAVING clause place conditions on groups in contrast to WHERE clause that place conditions on individual rows. While WHERE condition cannot include aggregate functions, HAVING conditions can do so.e.g.,

```
Select deptno,avg(sal), sum(sal) from emp group by deptno having deptno=10;
Select job, count(*) from emp group by job having count(*)<3;
```

7. **The DROP Command :** The DROP TABLE command is used to drop (delete) a table from database. But there is a condition for droping a table ; it must be an empty table i.e. a table with rows in it cannot be dropped.The syntax of this command is :

```
DROP TABLE <Table_name>;
```

e.g.,

```
DROP TABLE EMP;
```

8. **Query Based on Two table (Join):**

```
SELECT <Column-list>
FROM <table_name1>,<table_name2>
WHERE <Join_condition>[AND condition];
```

9. **Qualified Names :**

<tablename>.<fieldname>

This type of field names are called qualified field names and are used to identifying a field if the two joining tables have fields with same name.

<u>6 – Marks Questions</u>

Q1. Consider the following tables GAMES and PLAYER. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii).

Table: GAMES

GCode	GameName	Number	PrizeMoney	ScheduleDate
101	Carom Board	2	5000	23-Jan-2004
102	Badminton	2	12000	12-Dec-2003
103	Table Tennis	4	8000	14-Feb-2004

Downloaded from www.studiestoday.com

Downloaded from www.studiestoday.com

105	Chess	2	9000	01-Jan-2004
108	Lawn Tennis	4	25000	19-Mar-2004

Table: PLAYER

PCode	Name	Gcode
1	Nabi Ahmad	101
2	Ravi Sahai	108
3	Jatin	101
4	Nazneen	103

(i) To display the name of all Games with their Gcodes.
(ii) To display details of those games which are having PrizeMoney more than 7000.
(iii) To display the content of the GAMES table in ascending order of ScheduleDate.
(iv) To display sum of PrizeMoney for each of the Number of participation groupings (as shown in column Number 2 or 4).
(v) SELECT COUNT(DISTINCT Number) FROM GAMES;
(vi) SELECT MAX(ScheduleDate),MIN(ScheduleDate) FROM GAMES;
(vii) SELECT SUM(PrizeMoney) FROM GAMES;
(viii) SELECT DISTINCT Gcode FROM PLAYER;

Ans : (i) SELECT GameName,Gcode FROM GAMES;
(ii) SELECT * FROM GAMES WHERE PrizeMoney>7000;
(iii) SELECT * FROM GAMES ORDER BY ScheduleDate;
(iv) SELECT SUM(PrizeMoney),Number FROM GAMES GROUP BY Number;
(v) 2
(vi) 19-Mar-2004 12-Dec-2003
(vii) 59000
(viii)

```
101
103
108
```

Q2. Consider the following tables FACULTY and COURSES. Write SQL commands for the statements (i) to (v) and give outputs for SQL queries (vi) to (vii).

FACULTY

F_ID	Fname	Lname	Hire_date	Salary
102	Amit	Mishra	12-10-1998	12000
103	Nitin	Vyas	24-12-1994	8000
104	Rakshit	Soni	18-5-2001	14000
105	Rashmi	Malhotra	11-9-2004	11000
106	Sulekha	Srivastava	5-6-2006	10000

COURSES

C_ID	F_ID	Cname	Fees
C21	102	Grid Computing	40000
C22	106	System Design	16000
C23	104	Computer Security	8000
C24	106	Human Biology	15000
C25	102	Computer Network	20000
C26	105	Visual Basic	6000

i) To display details of those Faculties whose salary is greater than 12000.
ii) To display the details of courses whose fees is in the range of 15000 to 50000 (both

Downloaded from www.studiestoday.com

Downloaded from www.studiestoday.com

values included).
iii) To increase the fees of all courses by 500 of "System Design" Course.
iv) To display details of those courses which are taught by 'Sulekha' in descending order of courses.
v) Select COUNT(DISTINCT F_ID) from COURSES;
vi) Select Fname,Cname from FACULTY,COURSES where COURSES.F_ID =FACULTY.F_ID;

Ans.: (i) Select * from faculty where salary > 12000;
(ii) Select * from Courses.where fees between 15000 and 50000;
(iii) Update courses set fees = fees + 500 where Cname = "System Design";
(iv) Select * from faculty fac,courses cour where fac.f_id = cour.f_id and fac.fname = 'Sulekha'order by cname desc;
(v) 4
(vi)

Fname	Cname
Amit	Grid Computing
Rakshit	Computer Security
Rashmi	Visual Basic
Sulekha	Human Biology

2- Marks Questions

Define the following terms with example:

(i) DDL	(ii) DML	(iii) Primary Key	(iv) Candidate Key
(v) Alternet Key	(vi) Foreign Key	(vii) Cardinality of relation	(viii) Degree of relation
(ix) Relation	(x) Attribute	(xi) Tuple	

Downloaded from www.studiestoday.com