

CLASS- XII
COMPUTER SCIENCE WITH
PYTHON

CHAPTER- 14

More on SQL

CONSTRAINTS

A condition or check applicable on a field or set of fields.

- Types of constraints:

1. **NOT NULL**- It ensures that a column cannot have NULL value.
2. **DEFAULT**- It provides a default value for a column when none is specified.
3. **UNIQUE**- It ensures that all values in a column are different.
4. **CHECK**- It make sure that all values in a column satisfy certain criteria.
5. **PRIMARY KEY**- It used to uniquely identify a raw in a table.
6. **FOREIGN KEY**- It used to ensure referential integrity of data.

NOT NULL CONSTRAINT

Example- Create table Customer

```
(SID integer NOT NULL,  
Lname varchar (30) NOT NULL,  
Fname varchar (30) ) ;
```

DEFAULT CONSTRAINT

Example- Create table student

```
( SID integer,  
  Lname varchar (30) ,  
  Fname varchar (30) ,  
  Score DEFAULT 80 ) ;
```

UNIQUE CONSTRAINT

Example- Create table customer

(SID integer UNIQUE,

Lname varchar (30),

Fname varchar (30));

CHECK CONSTRAINT

Example – Create table customer

(SID integer CHECK (SID > 0),

Lname varchar (30),

Fname varchar (30));

PRIMARY KEY CONSTRAINT

Example – Create table customer

(SID integer NOT NULL PRIMARY KEY,

Lname varchar (30),

Fname varchar (30));

FOREIGN KEY CONSTRAINT

Examples - Create table order

(Oid integer,

Cust_SID integer,

Amount double,

Primary key (Oid)

Foreign Key (Cust_SID) references Customer (SID));

INSERTING DATA INTO ANOTHER TABLE

Syntax- Insert into table 1

Select columns from table 2

Where condition;

Example- Insert into branch 1

Select * from branch 2

Where gross > 7000.00 ;

MODIFYING DATA IN TABLES

Syntax- Update table-name

Set column-names (< values >)

Where condition;

Example- Update items

Set rol = 400, QOH = 700

Where icode < 'I040' ;

DELETING DATA FROM TABLES

Syntax- **Delete from table-name**

Where condition ;

Example- **Delete from employee**

Where eno = ' E101' ;

ALTERING TABLE

Syntax- **Alter table table-name Change Old col-name New-col-name datatype (size) ;**

Example- **Alter table customer Change Fname FirstName varchar (20);**

Syntax- **Alter table table-name Modify (col-name datatype (size)) ;**

Example- **Alter table emp Modify (Job char (30)) ;**

Syntax- **Alter table table-name Add (col-name datatype (size));**

Example- **Alter table emp Add (tel_no integer);**

DROPPING TABLES

Syntax- **Drop table table-name;**

Example- **Drop table items;**

NOTE : Delete command will delete all the data from table.

Drop command will delete the data including the table.

SQL JOINS

A query that fetches data from two or more tables whose records are joined with one another based on a condition.

Syntax- **Select column-names**

From table 1 , table 2

Where join condition ;

Example- **Select ename , loc**

From empl , dept

Where ename = 'ANOOP' AND empl.deptno = dept.deptno ;

TYPES OF JOINS

1. Cartesian Product – Without any join condition it returns all the records joined with all the other records of other table.

2. Equi Join – Joins two or more tables based on a condition using equality operator.

3. Inner Join – In this join only those rows are returned from both the tables that satisfy join condition. It matches records based on one or more columns.

4. Natural Join – It compares all the same names column in both tables.

5. Left Join – Select the rows from both left and right tables that are matched plus all rows from left table (table A).

6. Right Join - Select the rows from both left and right tables that are matched plus all rows from right table (table B).

INDEXES IN DATABASE

An index stores the sorted/ordered values within the index field & their location in the actual table.

1. Create index at the time of table creation with **CREATE TABLE** Command.

Example- Create table Players

```
(PlayerNo integer NOT NULL,  
Name char (15) NOT NULL,  
Phone char (15),  
Teamno char (4) NOT NULL,  
Primary Key (PlayerNo),  
INDEX Player_idx (Teamno) );
```

It will create a table of name Players & at same time it will also create an index namely Player_idx & on basis of field TeamNo it will take default order ascending for index field.

2. Create index on an already existing table.

Example- **Create unique INDEX Player_idx
ON Players (TeamNo);**

It will create unique index on the basis of field TeamNo on existing table named Players.

ORDER BY CLAUSE

Example- **Select * from data Order by marks ;**

Example- **Select * from data Order by section ASC, marks DESC ;**

NOTE : ASC- Ascending and DESC- Descending.

- **Ordering on the basis of expression**

Example- **Select rollno, name, grade, section, marks * .35 from data Where marks >70**

Order by section ASC , marks * 0.35 DESC;

- **Specifying Custom Sort Order**

Example- **Select * from data Order by FIELD (Project, 'Evaluated', 'Pending', 'Submitted', 'Assigned');**

AGGREGATE FUNCTIONS

1. **AVG ()**-Return the average of values.
2. **COUNT ()**- Returns the no. of rows in a result set.
3. **MAX ()**- Returns the maximum value.
4. **MIN ()**- Returns the minimum value.
5. **SUM ()**- Returns the sum of values.

QUERIES WITH AGGREGATE FUNCTION

1. **AVG()**- Select AVG (sal) “Average” from emp ;
2. **COUNT()**- Select COUNT (*) “Total” from emp ;
3. **MAX()**- Select MAX (sal) “Maximum Salary” from emp ;
4. **MIN()**- Select MIN (sal) “Minimum Salary” from emp ;
5. **SUM()**- Select SUM (sal) “Total Salary” from emp ;

GROUP BY Clause

The GROUP BY clause is used with SELECT statement and aggregate functions to group rows together by common column values.

Example-**Select job, count (*), sum (sal) From emp Group by job ;**

GROUP BY with HAVING clause

The HAVING clause is used with GROUP BY to satisfy the condition for group or aggregate. It can only be used with SELECT statement.

It is similar to WHERE clause but it is applied to group but where is applied to individual rows.

Example- **Select job , count (*) From emp Group by job Having count (*) < 3;**

NOTE : SELECT query can contain both WHERE & HAVING but WHERE clause must appear before GROUP BY clause whereas HAVING clause appear after it but before ORDER BY clause.