**SQL WHERE Clause**



SELECT ID, NAME, SALARYFROM CUSTOMERSWHERE SALARY >2000;

**SQL UPDATE Query**



UPDATE CUSTOMERS
SET ADDRESS = 'Pune'
WHERE ID = 6;

UPDATE CUSTOMERS
SET ADDRESS = 'Pune' ,SALARY = 1000.00;

**SQL DELETE Query**



DELETE FROM CUSTOMERSWHERE ID = 6;

DELETE FROM CUSTOMERS;

**SQL AND and OR Operators**



SELECT ID, NAME, SALARY FROM CUSTOMERS WHERE SALARY >2000 AND age <25;

**The OR Operator:**



SELECT ID, NAME, SALARY FROM CUSTOMERS WHERE SALARY >2000 OR age <25;

**SQL LIKE Clause**



SELECT \* FROM CUSTOMERSWHERE SALARY LIKE '200%' ;

**SQL ORDER BY Clause**



SELECT \* FROM CUSTOMERSORDER BY NAME, SALARY;

SELECT \* FROM CUSTOMERSORDER BY NAME DESC;

**SQL Group By**



SELECT NAME, SUM(SALARY) FROM CUSTOMERSGROUP BY NAME;

SELECT NAME, SUM(SALARY) FROM CUSTOMERSGROUP BY NAME;

**SQL Distinct Keyword**



SELECT SALARY FROM CUSTOMERSORDER BY SALARY;

SELECT DISTINCT SALARY FROM CUSTOMERSORDER BY SALARY;

**SQL SORTING Results**

SELECT \* FROM CUSTOMERSORDER BY NAME, SALARY;

SELECT \* FROM CUSTOMERS
ORDER BY NAME DESC;

**SQL Constraints:**

**NOT NULL Constraint:**

**Example:**

CREATE TABLE CUSTOMERS(ID INT NOT NULL,NAME VARCHAR (20) NOT NULL,AGE INT NOT NULL,ADDRESS CHAR (25) ,SALARY DECIMAL (18, 2),PRIMARY KEY (ID));

**DEFAULT Constraint:**

CREATE TABLE CUSTOMERS(ID INT NOT NULL,NAME VARCHAR (20) NOT NULL,AGE INT NOT NULL,ADDRESS CHAR (25) ,SALARY DECIMAL (18, 2) DEFAULT 5000.00,PRIMARY KEY (ID));

If CUSTOMERS table has already been created, then to add a DFAULT constraint to SALARY column, you wouldwrite a statement similar to the following:

ALTER TABLE CUSTOMERS
MODIFY SALARY DECIMAL (18, 2) DEFAULT 5000.00;

**Drop Default Constraint:**

ALTER TABLE CUSTOMERS
ALTER COLUMN SALARY DROP DEFAULT;

**UNIQUE Constraint:**

CREATE TABLE CUSTOMERS(ID INT NOT NULL,NAME VARCHAR (20) NOT NULL,AGE INT NOT NULL UNIQUE,ADDRESS CHAR (25) ,SALARY DECIMAL (18, 2),PRIMARY KEY (ID));

**DROP a UNIQUE Constraint:**

ALTER TABLE CUSTOMERS
DROP CONSTRAINT UNIQUE;

**PRIMARY Key:**

CREATE TABLE CUSTOMERS(ID INT NOT NULL,NAME VARCHAR (20) NOT NULL,AGE INT NOT NULL,ADDRESS CHAR (25) ,SALARY DECIMAL (18, 2),PRIMARY KEY (ID));

To create a PRIMARY KEY constraint on the "ID" column when CUSTOMERS table already exists, use thefollowing SQL syntax:

ALTER TABLE CUSTOMER ADD PRIMARY KEY (ID);

CREATE TABLE CUSTOMERS(ID INT NOT NULL,NAME VARCHAR (20) NOT NULL,AGE INT NOT NULL,ADDRESS CHAR (25) ,SALARY DECIMAL (18, 2),PRIMARY KEY (ID, NAME));

To create a PRIMARY KEY constraint on the "ID" and "NAMES" columns when CUSTOMERS table already exists,use the following SQL syntax:

**Delete Primary Key:**

ALTER TABLE CUSTOMERS DROP PRIMARY KEY ;

**FOREIGN Key:**

**CUSTOMERS table:**

CREATE TABLE CUSTOMERS(ID INT NOT NULL,NAME VARCHAR (20) NOT NULL,AGE INT NOT NULL,ADDRESS CHAR (25) ,SALARY DECIMAL (18, 2),PRIMARY KEY (ID));

**ORDERS table:**

CREATE TABLE ORDERS (ID INT NOT NULL,DATE DATETIME,CUSTOMER\_ID INT references CUSTOMERS(ID),AMOUNT double,PRIMARY KEY (ID));

If ORDERS table has already been created, and the foreign key has not yet been set, use the syntax for specifyinga foreign key by altering a table.

ALTER TABLE ORDERS
ADD FOREIGN KEY (Customer\_ID) REFERENCES CUSTOMERS (ID);

**DROP a FOREIGN KEY Constraint:**

ALTER TABLE ORDERS
DROP FOREIGN KEY;

**CHECK Constraint:**

CREATE TABLE CUSTOMERS(ID INT NOT NULL,NAME VARCHAR (20) NOT NULL,AGE INT NOT NULL CHECK (AGE >= 18),ADDRESS CHAR (25) ,SALARY DECIMAL (18, 2),PRIMARY KEY (ID));

If CUSTOMERS table has already been created, then to add a CHECK constraint to AGE column, you would writea statement similar to the following:

ALTER TABLE CUSTOMERS
MODIFY AGE INT NOT NULL CHECK (AGE >= 18 );

**DROP a CHECK Constraint:**

ALTER TABLE CUSTOMERS
DROP CONSTRAINT myCheckConstraint;

**INDEX:**

CREATE TABLE CUSTOMERS(ID INT NOT NULL,NAME VARCHAR (20) NOT NULL,AGE INT NOT NULL,ADDRESS CHAR (25) ,SALARY DECIMAL (18, 2),PRIMARY KEY (ID));

To create an INDEX on AGE column, to optimize the search on customers for a particular age, following is the SQLsyntax:

CREATE INDEX idx\_age
ON CUSTOMERS ( AGE );

**DROP an INDEX Constraint:**

ALTER TABLE CUSTOMERS
DROP INDEX idx\_age;

**SQL Comparison Operators:**

Consider the CUSTOMERS table having the following records:

SQL>SELECT \* FROM CUSTOMERS;



SELECT \* FROM CUSTOMERS WHERE SALARY >5000;

SELECT \* FROM CUSTOMERS WHERE SALARY = 2000;

SELECT \* FROM CUSTOMERS WHERE SALARY != 2000;

SELECT \* FROM CUSTOMERS WHERE SALARY <>2000;

SELECT \* FROM CUSTOMERS WHERE SALARY >= 6500;

**SQL Logical Operators:**





SELECT \* FROM CUSTOMERS WHERE AGE >= 25 AND SALARY >= 6500;

SELECT \* FROM CUSTOMERS WHERE AGE >= 25 OR SALARY >= 6500;

SELECT \* FROM CUSTOMERS WHERE AGE IS NOT NULL;

SELECT \* FROM CUSTOMERS WHERE NAME LIKE 'Ko%' ;

SELECT \* FROM CUSTOMERS WHERE AGE IN ( 25, 27 );

SELECT \* FROM CUSTOMERS WHERE AGE BETWEEN 25 AND 27;

SELECT AGE FROM CUSTOMERS
WHERE EXISTS (SELECT AGE FROM CUSTOMERS WHERE SALARY >6500);

SELECT \* FROM CUSTOMERS
WHERE AGE >ALL (SELECT AGE FROM CUSTOMERS WHERE SALARY >6500);

SELECT \* FROM CUSTOMERS
WHERE AGE >ANY (SELECT AGE FROM CUSTOMERS WHERE SALARY >6500);

**SQL - Boolean Expressions:**



SELECT \* FROM CUSTOMERS WHERE SALARY = 10000;

**SQL - Date Expressions:**

SELECT CURRENT\_TIMESTAMP;

**SQL Joins**

Consider the following two tables, (a) CUSTOMERS table is as follows:



(b) Another table is ORDERS as follows:



SELECT ID, NAME, AGE, AMOUNT
FROM CUSTOMERS, ORDERS
WHERE CUSTOMERS.ID = ORDERS. CUSTOMER\_ID;

**INNER JOIN**

SELECT ID, NAME, AMOUNT
FROM CUSTOMERS
INNER JOIN ORDERS
ON CUSTOMERS. ID = ORDERS. CUSTOMER\_ID;

**LEFT JOIN**

SELECT ID, NAME, AMOUNT
FROM CUSTOMERS
LEFT JOIN ORDERS
ON CUSTOMERS. ID = ORDERS. CUSTOMER\_ID;

**RIGHT JOIN**

SELECT ID, NAME, AMOUNT
FROM CUSTOMERS
RIGHT JOIN ORDERS
ON CUSTOMERS. ID = ORDERS. CUSTOMER\_ID;

**FULL JOIN**

SELECT ID, NAME, AMOUNT
FROM CUSTOMERS
FULL JOIN ORDERS
ON CUSTOMERS. ID = ORDERS. CUSTOMER\_ID;

**CARTESIAN JOIN**

SELECT ID, NAME, AMOUNT
FROM CUSTOMERS, ORDERS;

**The UNION ALL Clause:**

SELECT ID, NAME, AMOUNT
FROM CUSTOMERS
LEFT JOIN ORDERS
ON CUSTOMERS. ID = ORDERS. CUSTOMER\_ID
UNION ALL
SELECT ID, NAME, AMOUNT

FROM CUSTOMERS
RIGHT JOIN ORDERS
ON CUSTOMERS. ID = ORDERS. CUSTOMER\_ID;

**INTERSECT Clause**

SELECT ID, NAME, AMOUNT
FROM CUSTOMERS
LEFT JOIN ORDERS
ON CUSTOMERS. ID = ORDERS. CUSTOMER\_ID
INTERSECT
SELECT ID, NAME, AMOUNT
FROM CUSTOMERS
RIGHT JOIN ORDERS
ON CUSTOMERS. ID = ORDERS. CUSTOMER\_ID;

**EXCEPT Clause**

SELECT ID, NAME, AMOUNT
FROM CUSTOMERS
LEFT JOIN ORDERS
ON CUSTOMERS.ID = ORDERS.CUSTOMER\_ID
EXCEPT
SELECT ID, NAME, AMOUNT

FROM CUSTOMERS
RIGHT JOIN ORDERS
ON CUSTOMERS.ID = ORDERS.CUSTOMER\_ID;

**TRUNCATE TABLE**

TRUNCATE TABLE table\_name;

**Creating Views:**

Consider the CUSTOMERS table having the following records:





CREATE VIEW CUSTOMERS\_VIEW AS
SELECT name, age
FROM CUSTOMERS;

SELECT \* FROM CUSTOMERS\_VIEW;

**Updating a View:**

UPDATE CUSTOMERS\_VIEW
SET AGE = 35
WHERE name='Ramesh' ;

Deleting Rows into a View:

DELETE FROM CUSTOMERS\_VIEW
WHERE age = 22;

Dropping Views:

DROP VIEW CUSTOMERS\_VIEW;

SQL Transactions



The COMMIT Command:

DELETE FROM CUSTOMERS
WHERE AGE = 25;
COMMIT;

The ROLLBACK Command:

DELETE FROM CUSTOMERS
WHERE AGE = 25;
ROLLBACK;

The SAVEPOINT Command:

SAVEPOINT SP1;

DELETE FROM CUSTOMERS WHERE ID=1;

ROLLBACK TO SP2;