**SQL cheat sheet**

|  |  |  |
| --- | --- | --- |
| Data types | | |
| BLOB | BOOLEAN | CHAR |
| INT | FLOAT | TIME |
| TIMESTAMP | VARCHAR | DATE |

|  |  |  |  |
| --- | --- | --- | --- |
| Comparator operators | | | |
| = | Equal to | != | Not equal to |
| > | Greater than | >= | Greater than or equal to |
| < | Less than | <= | Less than or equal to |
| BETWEEN … AND … | Between two values | IN (list) | Any list of values |
| LIKE | Match a character pattern | IS NULL | Is a null value |

|  |  |  |  |
| --- | --- | --- | --- |
| Commands | | | |
| CREATE | Make tables | DELETE | Remove data |
| INSERT (INTO) | Add data | UPDATE | Modify data |
| SELECT | View data | DROP | Destroy table |

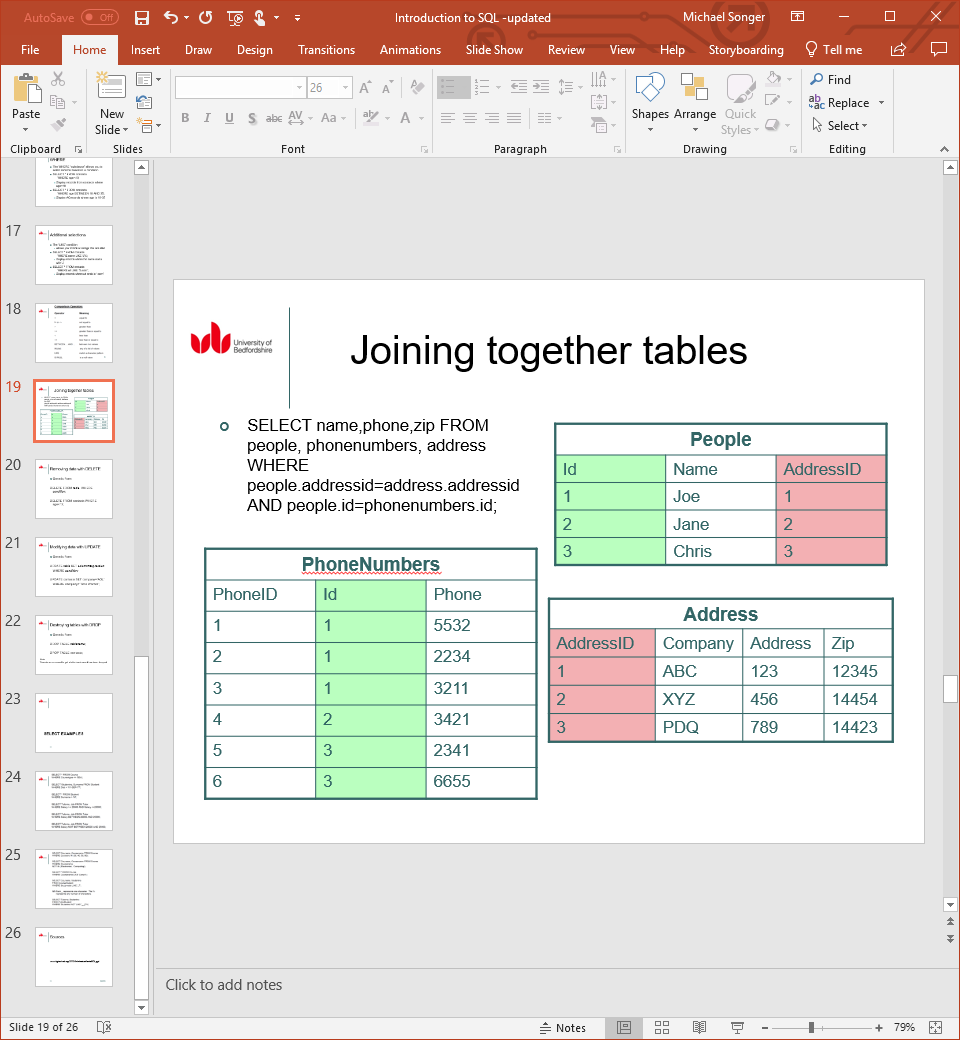
|  |  |
| --- | --- |
| CREATE examples | |
| CREATE TABLE tablename (  Column\_name data\_type attributes); | CREATE TABLE contacts (  ContactID INT PRIMARY KEY,  Name VARCHAR(40),  Address VARCHAR(60),  Company VARCHAR(60),  Phone VARCHAR(11),  URL VARCHAR(80),  age INT,  Height FLOAT,  Birthday DATE,  WhenEntered TIMESTAMP  ); |
| CREATE TABLE Orders (     OrderID int NOT NULL,     OrderNumber int NOT NULL,     PersonID int,     PRIMARY KEY (OrderID),     FOREIGN KEY (PersonID) REFERENCES Persons(PersonID) ); |  |

|  |  |
| --- | --- |
| DELETE examples | |
| DELETE FROM table WHERE condition; | DELETE FROM contacts WHERE age<13; |

|  |  |
| --- | --- |
| UPDATE examples | |
| UPDATE table SET column = expression  WHERE condition; | UPDATE contacts SET company = ‘AOL’  WHERE company = ‘Time Warner’; |

|  |  |
| --- | --- |
| DROP examples | |
| DROP TABLE tablename; | DROP TABLE contacts; |

|  |  |  |
| --- | --- | --- |
| SELECT Examples | | |
| SELECT column, … FROM table, …  WHERE condition  GROUP BY group by expression  HAVING condition  ORDER BY order expression | SELECT contactid, name  FROM contacts;  Display only the record number and names | SELECT DISTINCT url  FROM contacts;  Display only one entry for every value of URL. |
| SELECT \*  FROM contacts  WHERE name LIKE ‘J%’;  Display records where the name starts with ‘J’ | SELECT \*  FROM contacts  WHERE url LIKE ‘%.com’;  Display records where url ends in “.com” | SELECT \*  FROM contacts  WHERE age<10;  Display records from contacts where age<10 |
| SELECT \*  FROM contacts  WHERE age BETWEEN 18 AND 35;  Display all records where age is 18-35 | SELECT \*  FROM contacts WHERE 1;  Displays all contacts | SELECT Courseno, Studentno  FROM CourseStudent  WHERE Studymode  LIKE ‘\_T’;  \_ represents one character % represents any number of characters |
| SELECT \* FROM Course  WHERE CourseType <> ‘BSc’; |  |  |



|  |  |  |
| --- | --- | --- |
| INSERT examples | | |
| INSERT INTO tablename  (column\_name, …)  VALUES (value, …); | INSERT INTO contacts (contactid, name, address, company, phone, url, age, height, birthday, whenentered)  VALUES (1, ‘Joe’, ’123 Any St.’, ’ABC’, ’800-555-1212’, ‘http://abc.com’, 30, 1.9, ’6/14/1972’, now()); | INSERT INTO contacts (contactid, name, phone)  VALUES (2,’Jane’,’212-555-1212’); |

CONSTRAINTS

|  |
| --- |
| Column constraint level (applies to a single column)  column [CONSTRAINT constraint\_name] constraint\_type, |
| CREATE TABLE emp(  empno INT,  ename VARCHAR(10) NOT NULL,  job VARCHAR(9),  mgr INT,  hiredate DATE,  sal FLOAT(7,2),  comm FLOAT(7,2),  deptno FLOAT(7,2) NOT NULL  ); |

|  |  |
| --- | --- |
| table constraint level (applies to one or more columns)  column, ...  [CONSTRAINT constraint\_name constraint\_type (column, …), | |
| CREATE TABLE emp(  empno INT,  ename VARCHAR(10) NOT NULL,  job VARCHAR(9),  mgr INT,  hiredate DATE,  sal FLOAT(7,2),  comm FLOAT(7,2),  deptno FLOAT(7,2) NOT NULL,  CONSTRAINT emp\_empno\_pk PRIMARY KEY(empno)  ); | CREATE TABLE dept(  deptno INT,  dname varchar(14),  loc varchar(13),  CONSTRAINT dept\_dname\_uk UNIQUE(dname),  CONSTRAINT dept\_dname\_pk PRIMARY KEY(deptno)  ); |
| CREATE TABLE emp (  empno INT,  Ename VARCHAR(10) NOT NULL,  Job VARCHAR(9),  Mgr INT,  Hiredate DATE,  Sal FLOAT(7,2),  Comm FLOAT(7,2),  Deptno FLOAT(7,2) NOT NULL,  CONSTRAINT emp\_deptno\_fk FOREIGN KEY (deptno)  REFERENCES dept (deptno)  ); |  |

ALTER TABLE Student

ADD CONSTRAINT Student\_PK

PRIMARY KEY (Studentno);

ALTER TABLE TutorStudent

ADD CONSTRAINT TutorStudent\_Tutorno\_FK

FOREIGN KEY (Tutorno)

REFERENCES Tutor (Tutorno);

Remove manager constraint from the EMP table

ALTER TABLE emp

DROP CONSTRAINT emp\_mgr\_fk;

Remove the manager constraint from the EMP table

ALTER TABLE emp

DROP CONSTRAINT emp\_mgr\_fk;

Remove the PRIMARY KEY constraint on the DEPT table and drop the associated FOREIGN KEY constraint on the EMP.DEPTNO

ALTER TABLE dept

DROP PRIMARY KEY CASCADE;

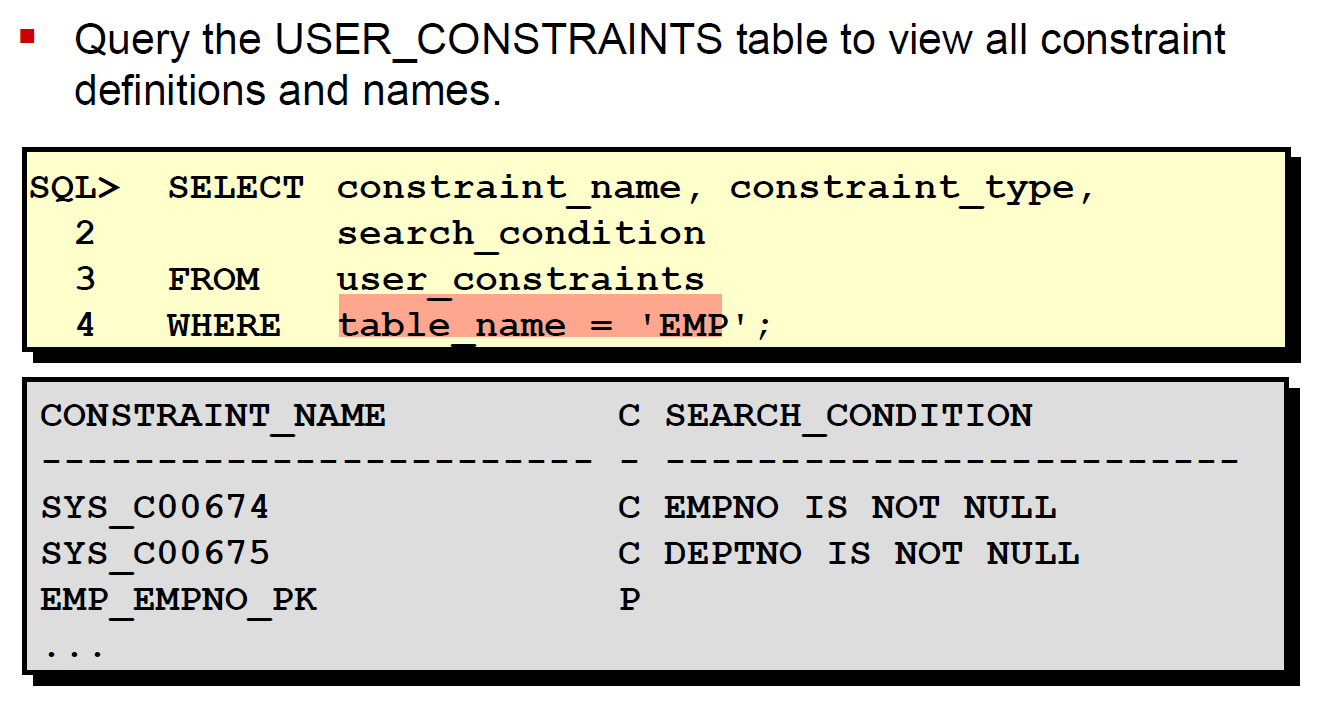
Query the USER\_CONSTRAINTS table to view all constraint definitions and names

SELECT constraint\_name, constraint\_type,

Search\_conidition

FROM user\_constraints

WHERE table\_name = ‘EMP’;



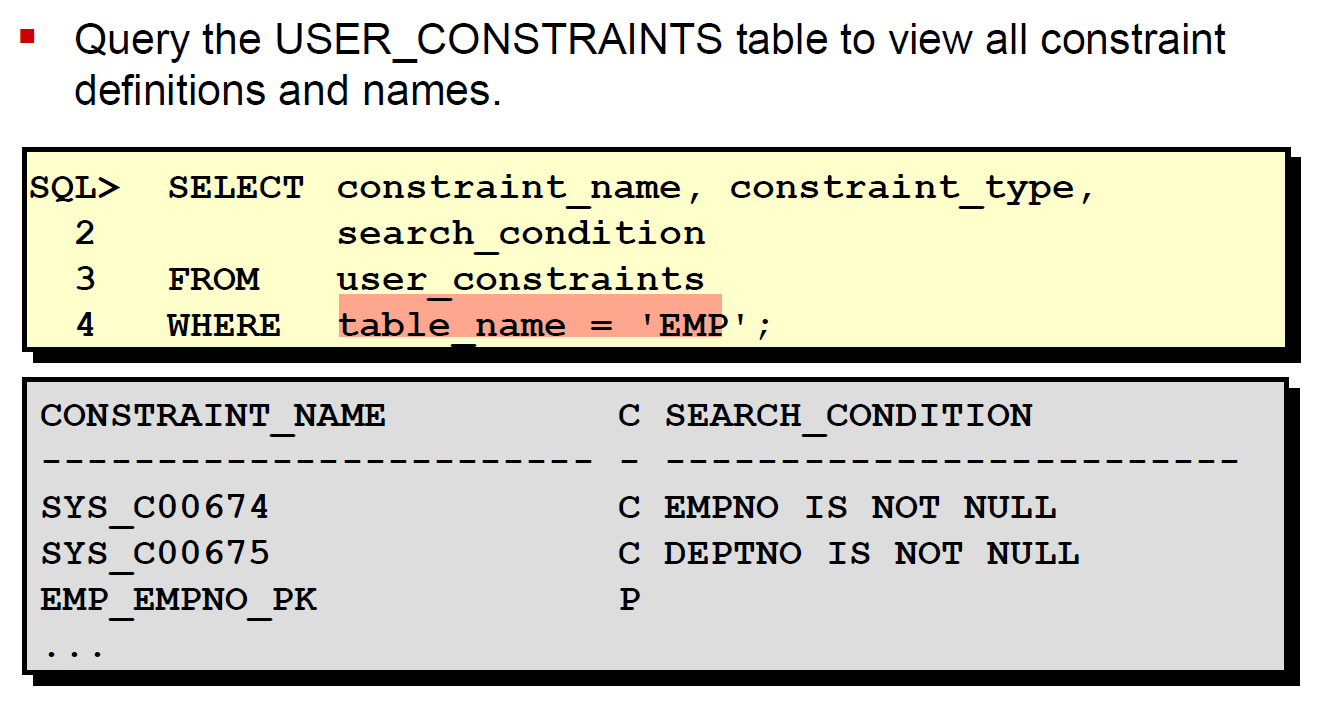
Query the USER\_CONSTRAINS ttable to view all constraint definitions and names.

SELECT constraint\_name, constraint\_type,

Search\_conidition

FROM user\_condition

WHERE table\_name = ‘EMP’;

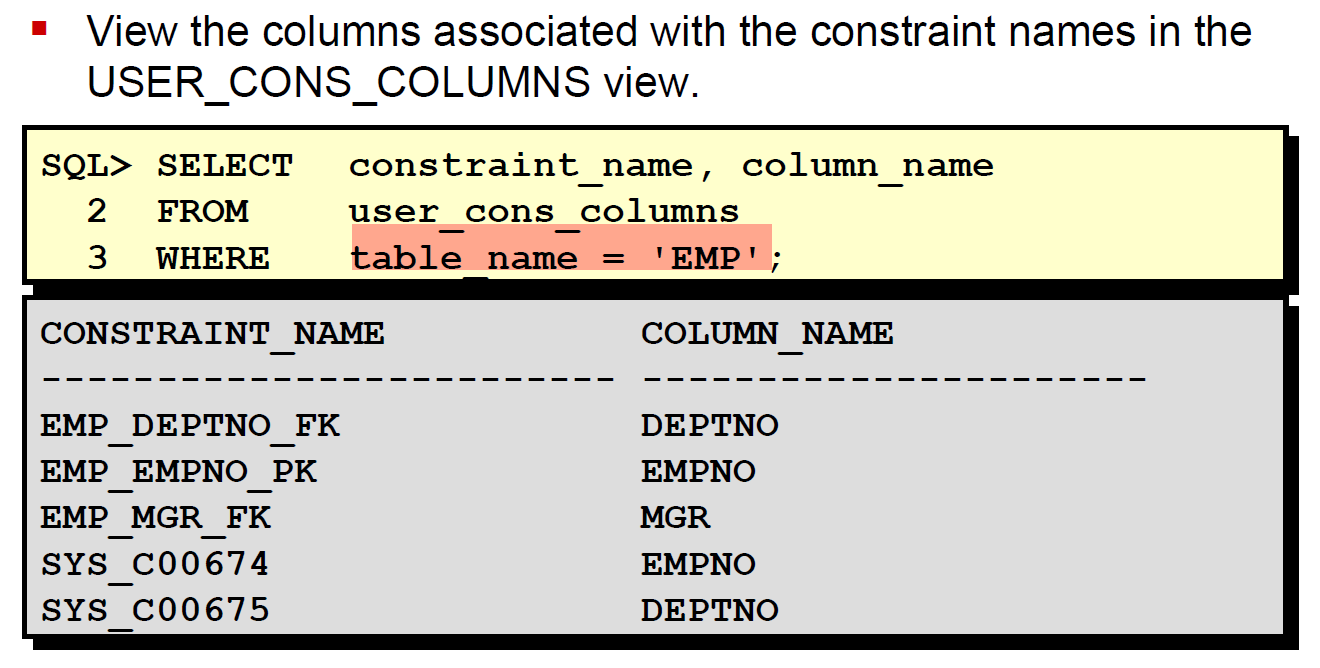


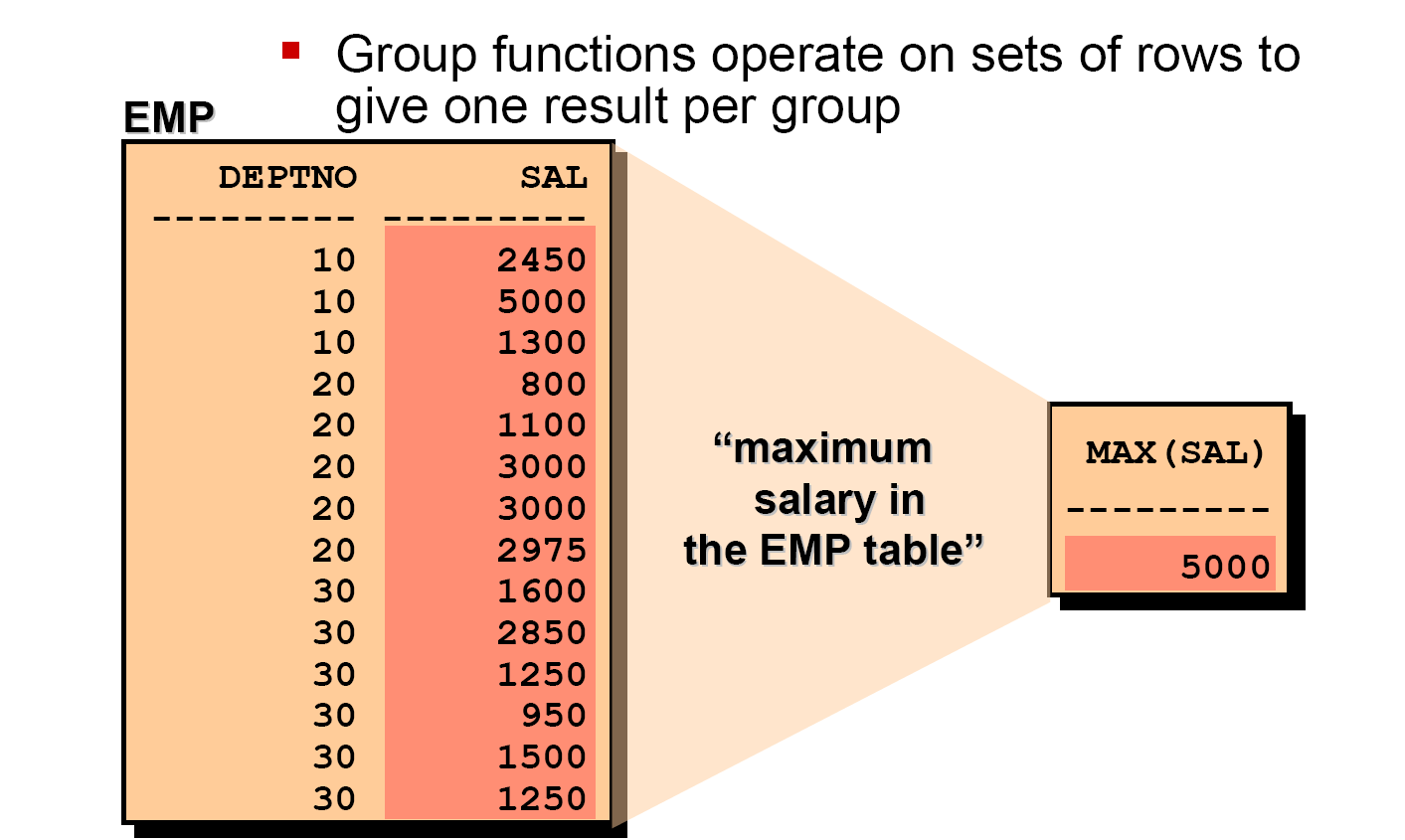
View the columns associated with the constraint names in the USER\_CONS\_COLUMNS view

SELECT constraint\_name, column\_name

FROM user\_cons\_columns

WHERE table\_name = ‘EMP’;





|  |  |  |
| --- | --- | --- |
| Group functions | | |
| AVG | COUNT | MAX |
| MIN | STDDEV | SUM |
| VARIANCE |  |  |

SELECT AVG(sal), MAX(sal),

MIN(sal), SUM(sal)

FROM emp

WHERE job LIKE ‘SALES%’;

SELECT job, SUM(sal) PAYROLL

FROM emp

WHERE job NOT LIKE ‘SALES%’

GROUP BY job

HAVING SUM(sal)>5000

ORDER BY SUM(sal);