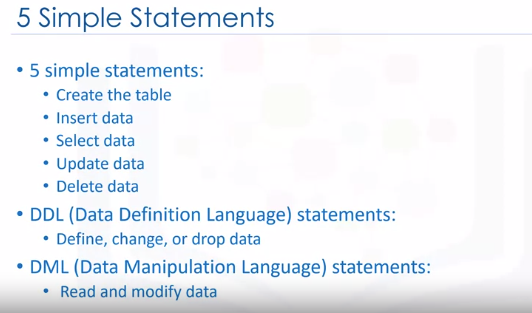
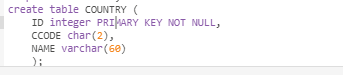


Three types of databases – relational databases, hierarchal, NOSQL

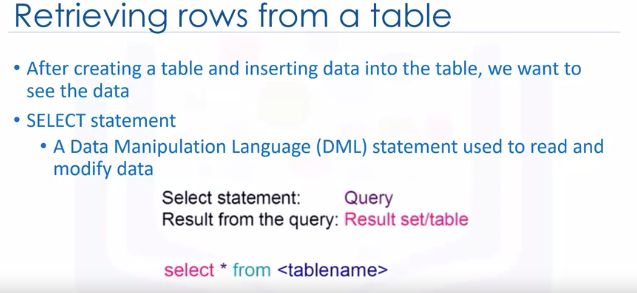


Create table statement





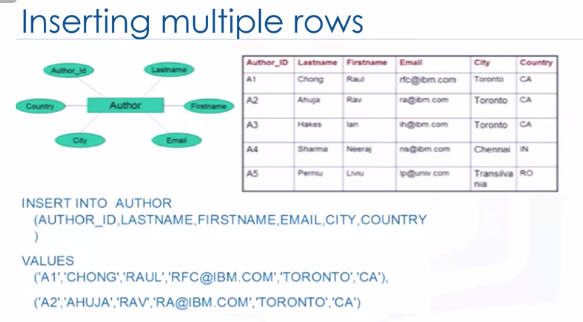
Drop table Example: - drop table COUNTRY;



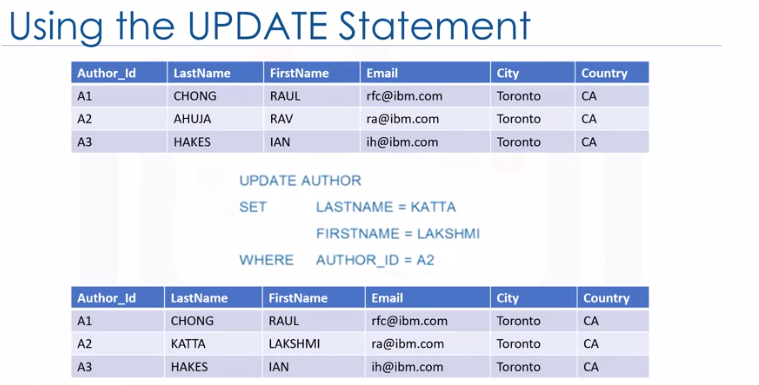
**select ID, NAME from COUNTRY where ID < 5 ;**

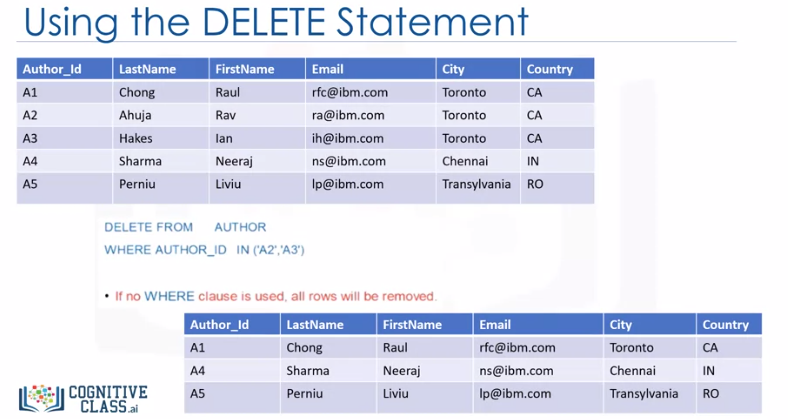
**SQL – COUNT, DISTINCT, LIMIT**

**Entity – attributes**

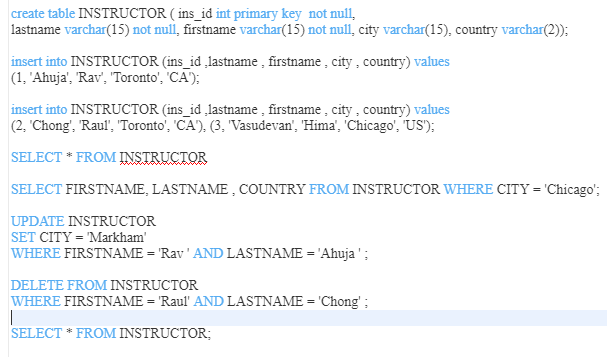


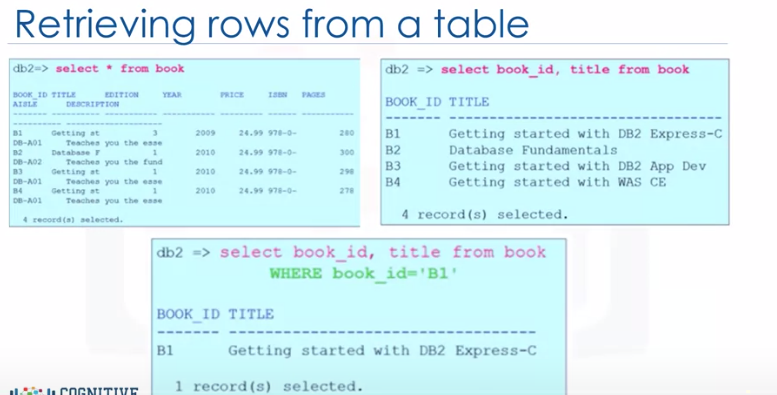
**Update statement**

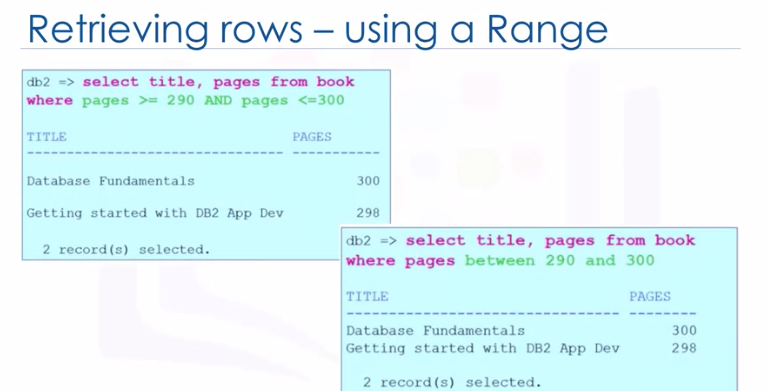


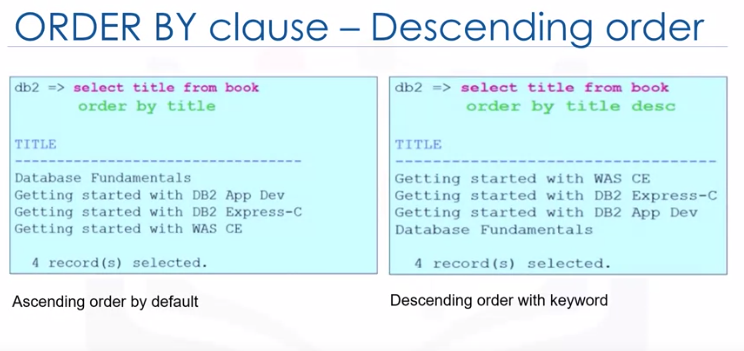


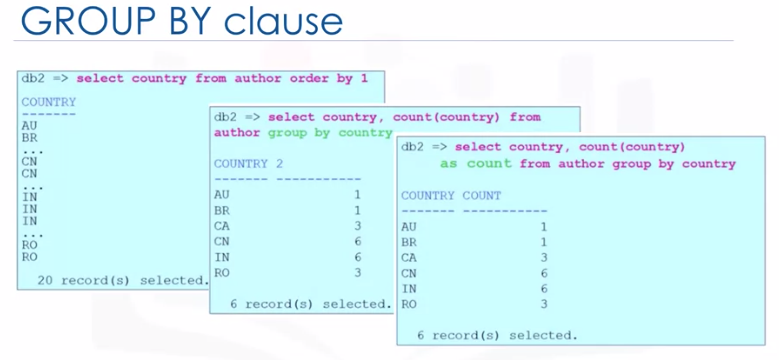
**EXAMPLE**

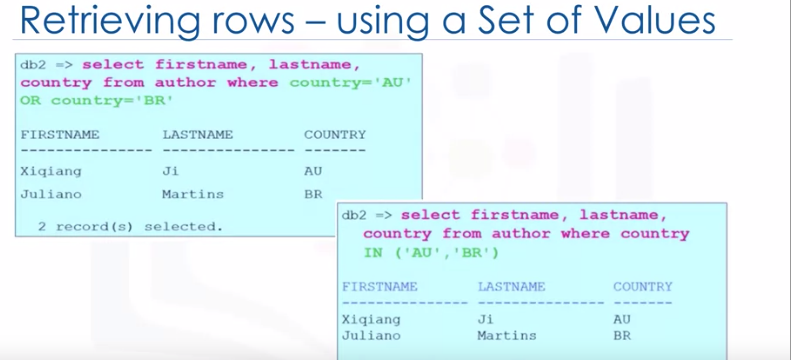


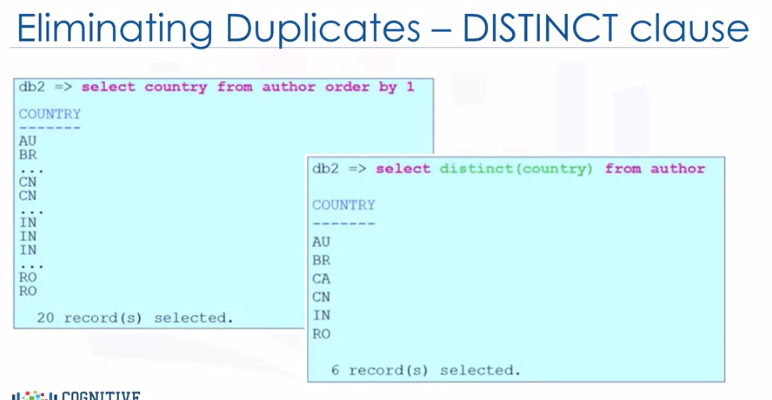


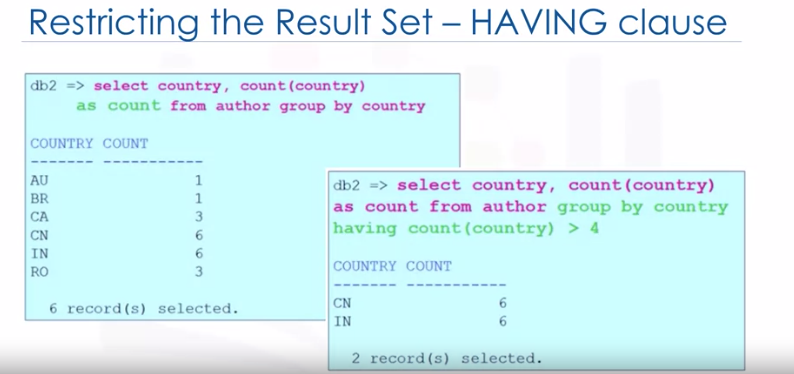


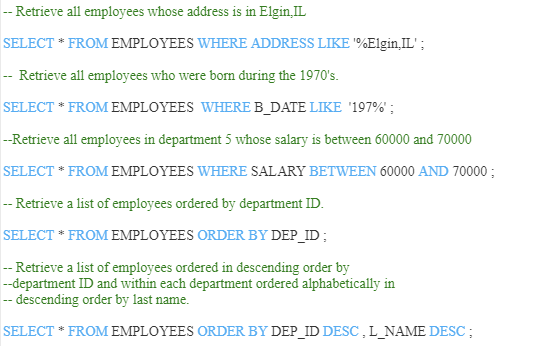


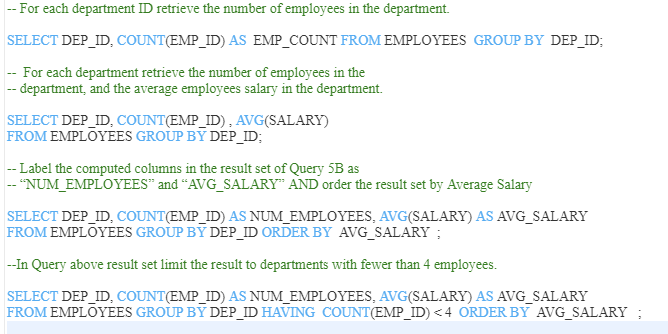












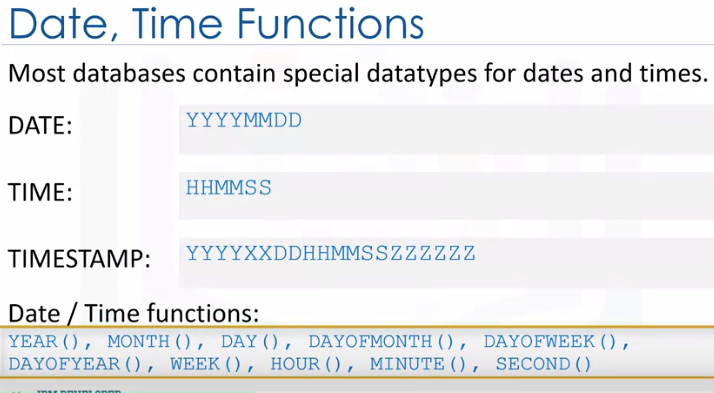
**SELECT DEP\_ID, COUNT(EMP\_ID) AS NUM\_EMPLOYEES,**

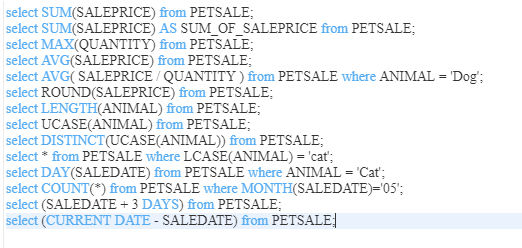
**AVG(SALARY) AS AVG\_SALARY**

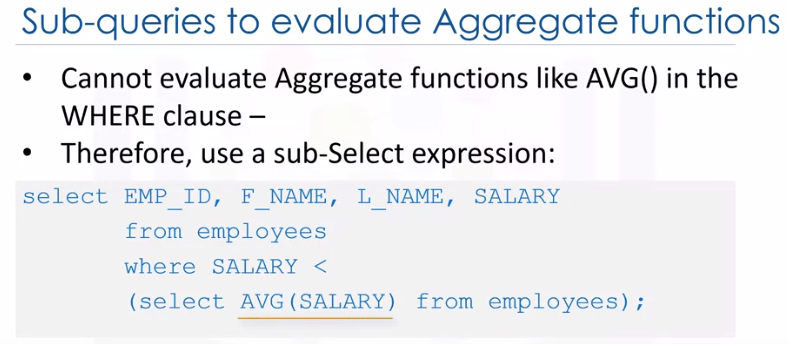
**FROM EMPLOYEES GROUP BY DEP\_ID**

**HAVING COUNT(EMP\_ID) < 4**

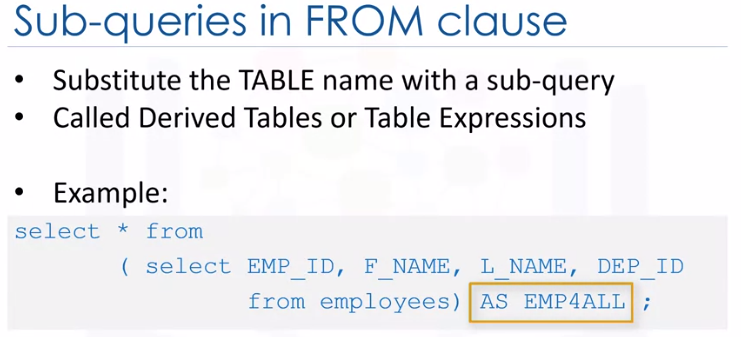
**ORDER BY AVG\_SALARY ;**

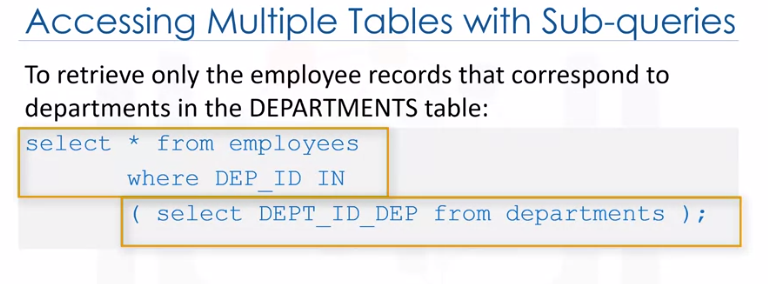


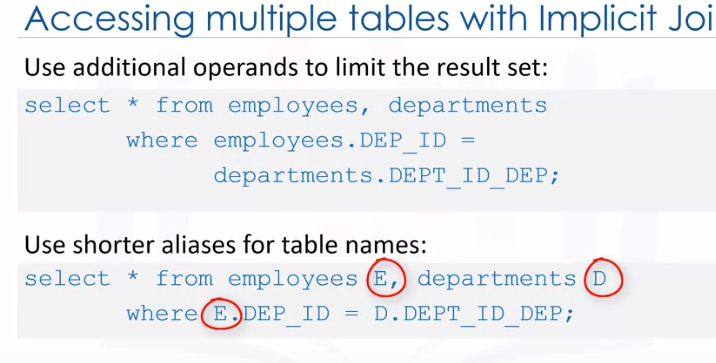












**SELECT E.EMP\_ID ,E.SALARY, D.DEP\_NAME FROM EMPLOYEES E, DEPARTMENTS D**

**WHERE E.DEP\_ID = D.DEPT\_ID\_DEP AND SALARY > (SELECT AVG(SALARY) FROM EMPLOYEES);**

**Example of Multiple Table & Sub Query**

**Select employee id , salary and dep\_name from emp & dep table**

**Where employee table\_dep\_id = dep table\_dep\_id and filter with avg salary subquery**

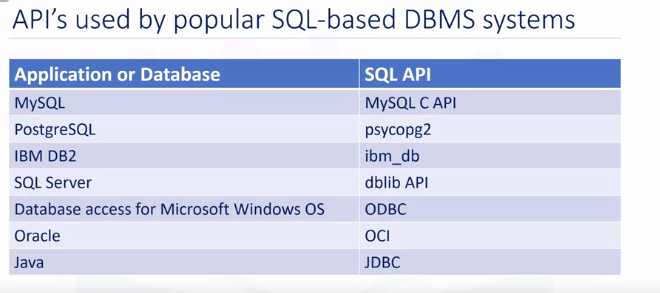
**SELECT E.EMP\_ID ,E.SALARY, D.DEP\_NAME FROM**

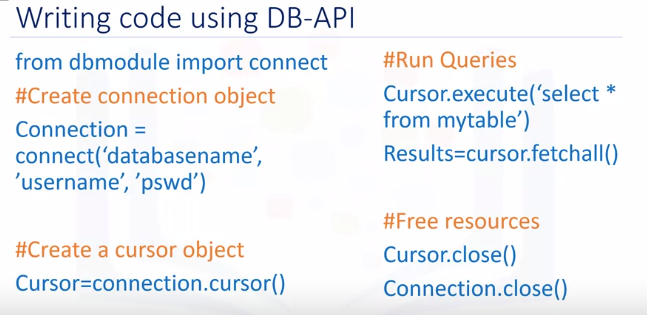
**EMPLOYEES E, DEPARTMENTS D**

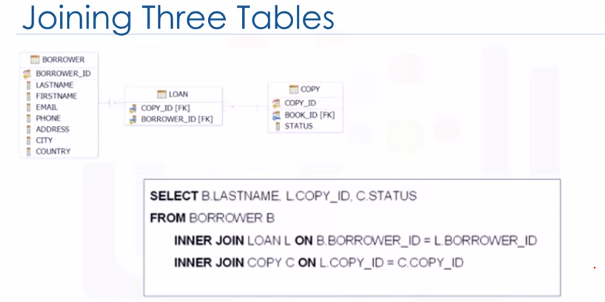
**WHERE E.DEP\_ID = D.DEPT\_ID\_DEP**

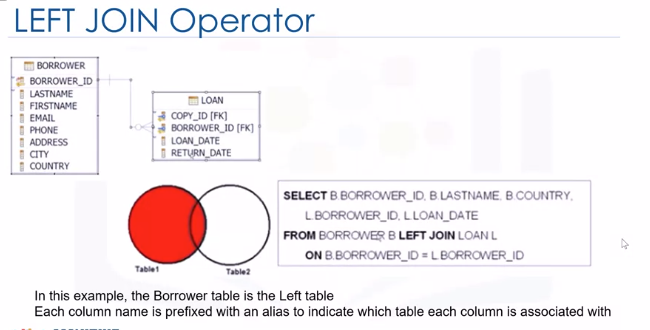
**AND SALARY > (SELECT AVG(SALARY) FROM EMPLOYEES);**

**select F\_NAME , DEP\_NAME FROM EMPLOYEES , DEPARTMENTS WHERE DEP\_ID = DEPT\_ID\_DEP ;**

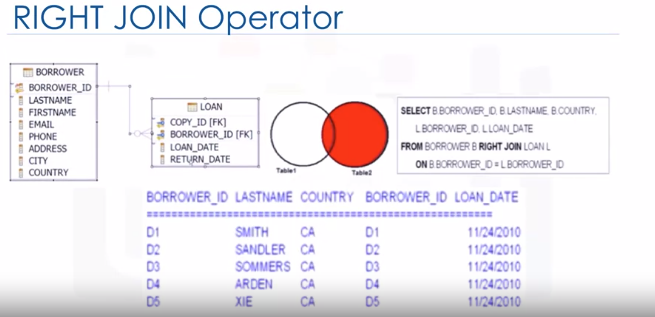


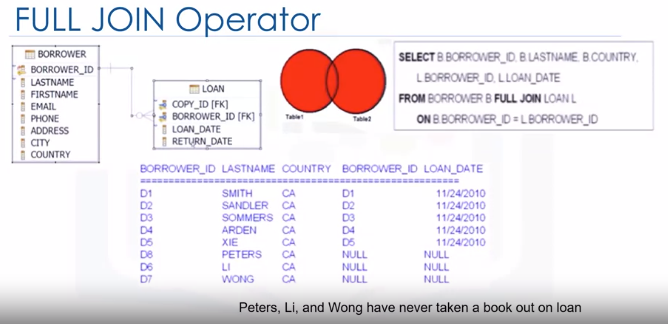




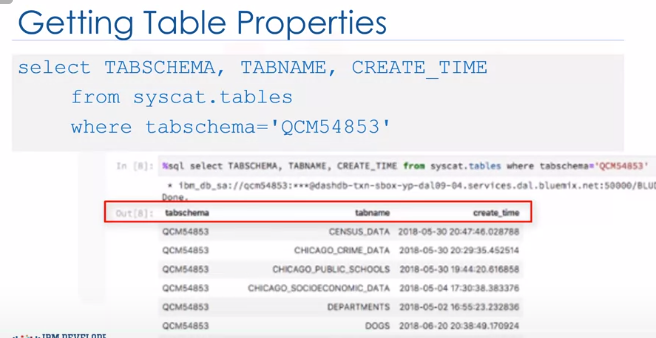




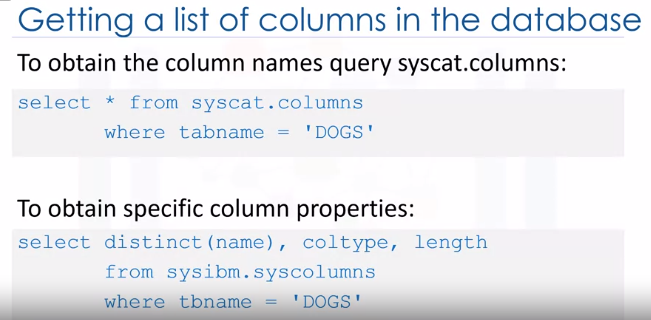


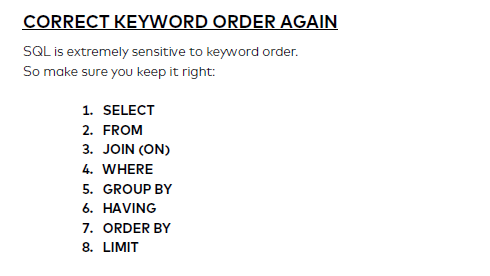


**Access Database Tables**



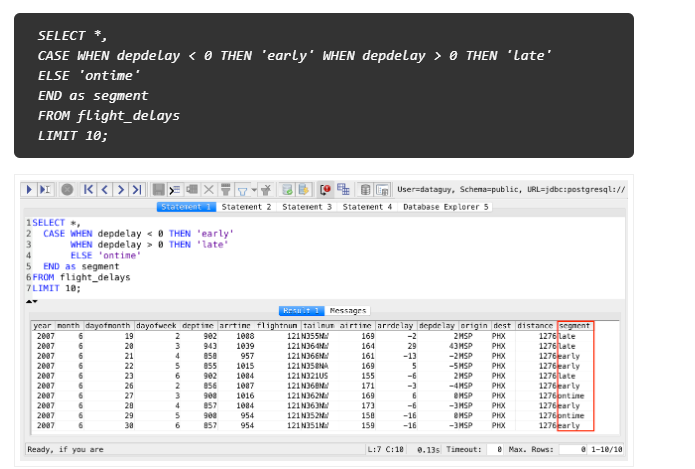
**Access Database Table - Columns**





SQL first checks which data table we will work with. Then it checks the filters. After that it groups the data. Finally it gets the data – and if necessary, sorts it and prints only the first X lines.

Data transformation – Feature engineering in SQL



SELECT COUNT(\*),

CASE WHEN depdelay < 0 THEN 'early'

WHEN depdelay > 0 THEN 'late'

ELSE 'ontime'

END as segment

FROM flight\_delays

GROUP BY segment

LIMIT 10;

some quick and dirty data cleaning/transformation

* change all the values from ‘M’ to ‘Male’ and from ‘F’ to ‘Female’ in a table
* turn a continuous value into a categorical value (like we did in the above example).