RELATIONAL DATABASES

GROUP BY

- IF YOU WANTED TO KNOW THE AVERAGE HEIGHT OF ALL STUDENTS WHAT SQL WOULD YOU WRITE?
- WHAT IF YOU WANTED TO KNOW THE AVERAGE HEIGHT OF THE STUDENTS FOR EACH YEAR IN THE SCHOOL?

GROUP BY

- TO SIMPLIFY THIS WE CAN WRITE JUST ONE STATEMENT WITH THE GROUP BY AND HAVING CLAUSE
- YOU USE THE GROUP BY CLAUSE TO DIVIDE THE ROWS IN A TABLE INTO SMALLER GROUPS.
- YOU CAN THEN USE THE GROUP FUNCTIONS TO RETURN SUMMARY INFORMATION FOR EACH GROUP.

GROUP BY

```
SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id
ORDER BY department_id;
```

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
50	3500
60	6400
80	10033.3333333333333333
90	19333.3333333333333333
110	10150
-	7000

- IN THE SELECT STATEMENT, THE ROWS ARE BEING GROUPED BY DEPARTMENT_ID
- THE AVG FUNCTION IS THEN APPLIED TO EACH GROUP

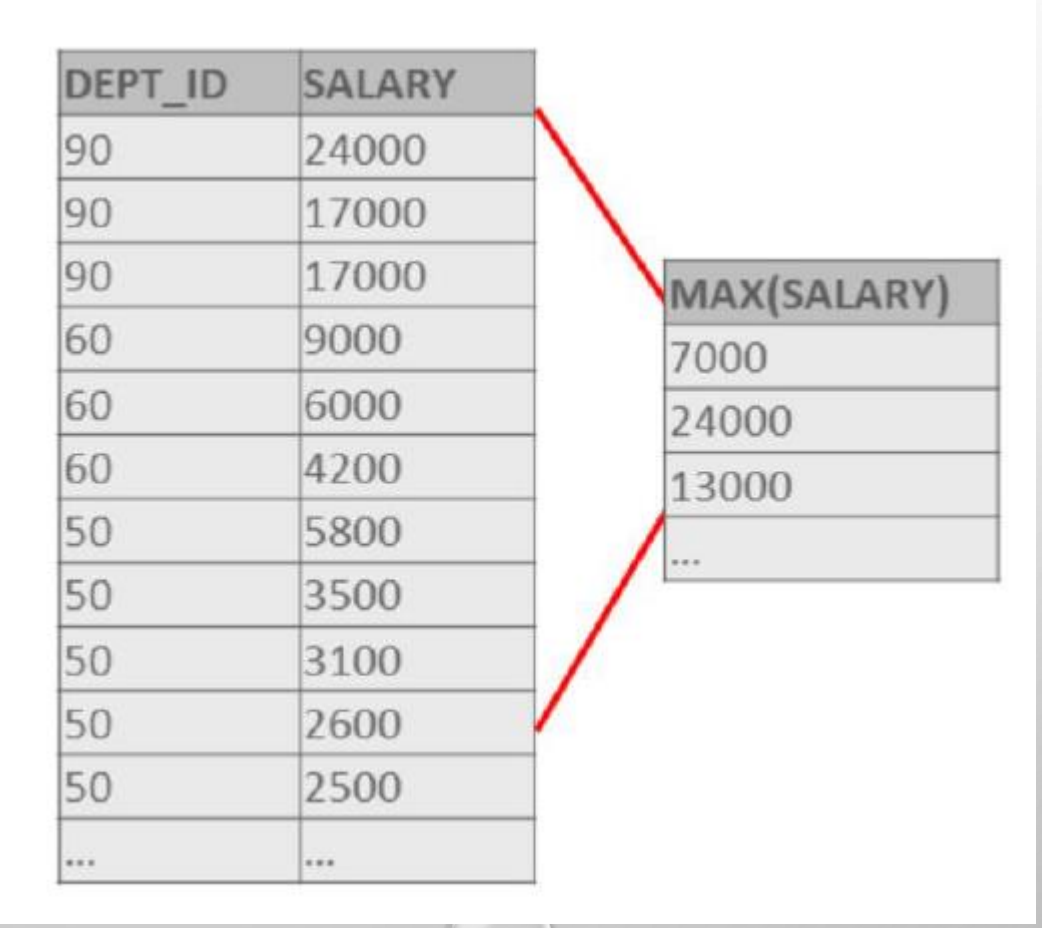
GROUP BY

- WHAT IF WE WANTED TO FIND THE MAXIMUM SALARY OF EMPLOYEES IN EACH DEPARTMENT?
- WE USE A GROUP BY CLAUSE STATING WHICH COLUMN TO USE TO GROUP THE ROWS.

```
SELECT MAX(salary)
FROM employees
GROUP BY department_id;
```

GROUP BY

- BUT HOW CAN WE TELL WHICH MAXIMUM SALARY BELONGS TO WHICH DEPARTMENT?



DEPT_ID	SALARY
90	24000
90	17000
90	17000
60	9000
60	6000
60	4200
50	5800
50	3500
50	3100
50	2600
50	2500
...	...

MAX(SALARY)
7000
24000
13000
...

GROUP BY

- USUALLY WE WANT TO INCLUDE THE GROUP BY COLUMN IN THE SELECT CLAUSE.

```
SELECT department_id, MAX(salary)
FROM employees
GROUP BY department_id;
```

DEPT_ID	SALARY
90	24000
90	17000
90	17000
60	9000
60	6000
60	4200
...	...

DEPT_ID	MAX(SALARY)
-	7000
90	24000
20	13000
...	...

GROUP BY

- GROUP FUNCTIONS REQUIRE THAN ANY COLUMN LISTED IN THE SELECT CLAUSE THAT IS NOT PART OF A GROUP FUNCTION MUST BE LISTED IN A GROUP BY CLAUSE.

```
SELECT job_id, last_name, AVG(salary)
FROM employees
GROUP BY job_id;
```

ORA-00979: not a GROUP BY expression

GROUP BY

- THIS EXAMPLE SHOWS HOW MANY COUNTRIES ARE IN EACH REGION.
- GROUP FUNCTIONS IGNORE NULLS SO IF ANY COUNTRY DOES NOT HAVE A COUNTRY NAME IT WILL NOT BE INCLUDED IN THE COUNT.

```
SELECT COUNT(country_name), region_id
FROM wf_countries
GROUP BY region_id
ORDER BY region_id;
```

COUNT(COUNTRY_NAME)	REGION_ID
15	5
28	9
21	11
8	13
7	14
8	15
5	17
17	18

GROUP BY

- WE CAN ALSO USE THE WHERE CLAUSE TO EXCLUDE ROWS BEFORE THE REMAINING ROWS ARE FORMED INTO GROUPS.

```
SELECT department_id, MAX(salary)
FROM employees
WHERE last_name != 'King'
GROUP BY department_id;
```

GROUP BY

LAST_NAME	DEPT_ID	SALARY
King	90	24000
Kochhar	90	17000
De Haan	90	17000
Hunold	60	9000
Ernst	60	6000
Lorentz	60	4200
...	...	...

DEPT_ID	MAX(SALARY)
-	7000
90	17000
20	13000
...	...

- EMPLOYEE KING IS EXCLUDED IN THE WHERE CLAUSE, THE MAX SALARY FOR DEPARTMENT 90 IS 17,000

GROUP BY

- WHAT IS THE AVERAGE POPULATION OF ALL COUNTRIES IN EACH REGION?
- COUNT THE NUMBER OF LANGUGAES FOR ALL COUNTRIES

- COUNTRIES TABLE HAS REGION_ID AND POPULATION
- LANGUAGES HAS LANGAGE_ID AND COUNTRY_ID

GROUP BY

- IF YOU INCLUDE A GROUP FUNCTION (AVG,SUM,COUNT,MAX,MIN,STDDEV,VARIANCE) IN A SELECT ALONG WITH ANY OTHER INDIVIDUAL COLUMNS, EACH OTHER COLUMN MUST APPEAR IN THE GROUP BY CLAUSE.
- YOU CANNOT USE A COLUMN ALIAS IN THE GROUP BY CLAUSE
- THE WHERE CLAUSE EXCLUDES ROWS BEFORE THEY ARE DIVIDED INTO GROUPS.

GROUP BY

- SOMETIMES YOU NEED TO DIVIDE GROUPS INTO SMALLER GROUPS
- YOU MAY WANT TO GROUP ALL EMPLOYEES BY DEPARTMENT, THEN WITHIN EACH DEPARTMENT GROUP THEM BY JOB, TO SHOW HOW MANY EMPLOYEES ARE DOING EACH JOB IN EACH DEPARTMENT

```
SELECT department_id, job_id, count(*)
FROM employees
WHERE department_id > 40
GROUP BY department_id, job_id;
```

DEPT_ID	JOB_ID	COUNT(*)
110	AC_ACCOUNT	1
50	ST_CLERK	4
80	SA_REP	2
90	AD_VP	2
50	ST_MAN	1
...	...	...

GROUP BY

- GROUP FUNCTIONS CAN BE NESTED TO A DEPTH OF TWO WHEN GROUP BY IS USED.

```
SELECT MAX(AVG(SALARY))
FROM EMPLOYEES
GROUP BY DEPARTMENT_ID;
```

HOW MANY VALUES WILL BE RETURNED BY THIS QUERY?