

Database System

Lecture 19

SQL Sub Languages

DRL Data Retrieval Language

DRL command

SELECT Statement

Group By clause

Using Aggregate Functions in Queries

Prepared By

Dhafer Sabah Yaseen

SQL Components Or SQL Sub Languages

DCL: Data Control Language

Example: Grant, Revoke.

DDL: Data Definition Language.

Example: Create, Alter, Drop, Rename and Truncate.

DML: Data Manipulation Language

Example: Insert, Update, Delete

DRL: Data Retrieval Language

Example: Select

TCL: Transaction Control Language

Example : Rollback, Commit, Savepoint

DRL-Data Retrieval Language

SELECT Statement :

Use a SELECT statement or subquery to retrieve data from one or more tables, object tables, views, object views.

Prerequisites

For you to select data from a table, view, object view, the object must be in your own schema or you must have the READ or SELECT privilege .

DRL-Data Retrieval Language

The Full syntax:

```
SELECT [DISTINCT]{* | {specific column}[[AS]c_alias]
        [, {specific column}[[AS] c_alias] ] ... }
FROM [schema.]{table | view }[t_alias]
     [, [schema.]{table | view }[t_alias] ] ...
[WHERE condition]
[GROUP BY expr[, expr] ...]
[HAVING condition]]
[{UNION | UNION ALL | INTERSECT | MINUS}
  SELECT command]
[ORDER BY {expr | position | c_alias } [ASC | DESC]
 [ NULLS FIRST | NULLS LAST ]
     [, {expr | position | c_alias } [ASC | DESC]
 [ NULLS FIRST | NULLS LAST ]...]
```

DRL-Data Retrieval Language

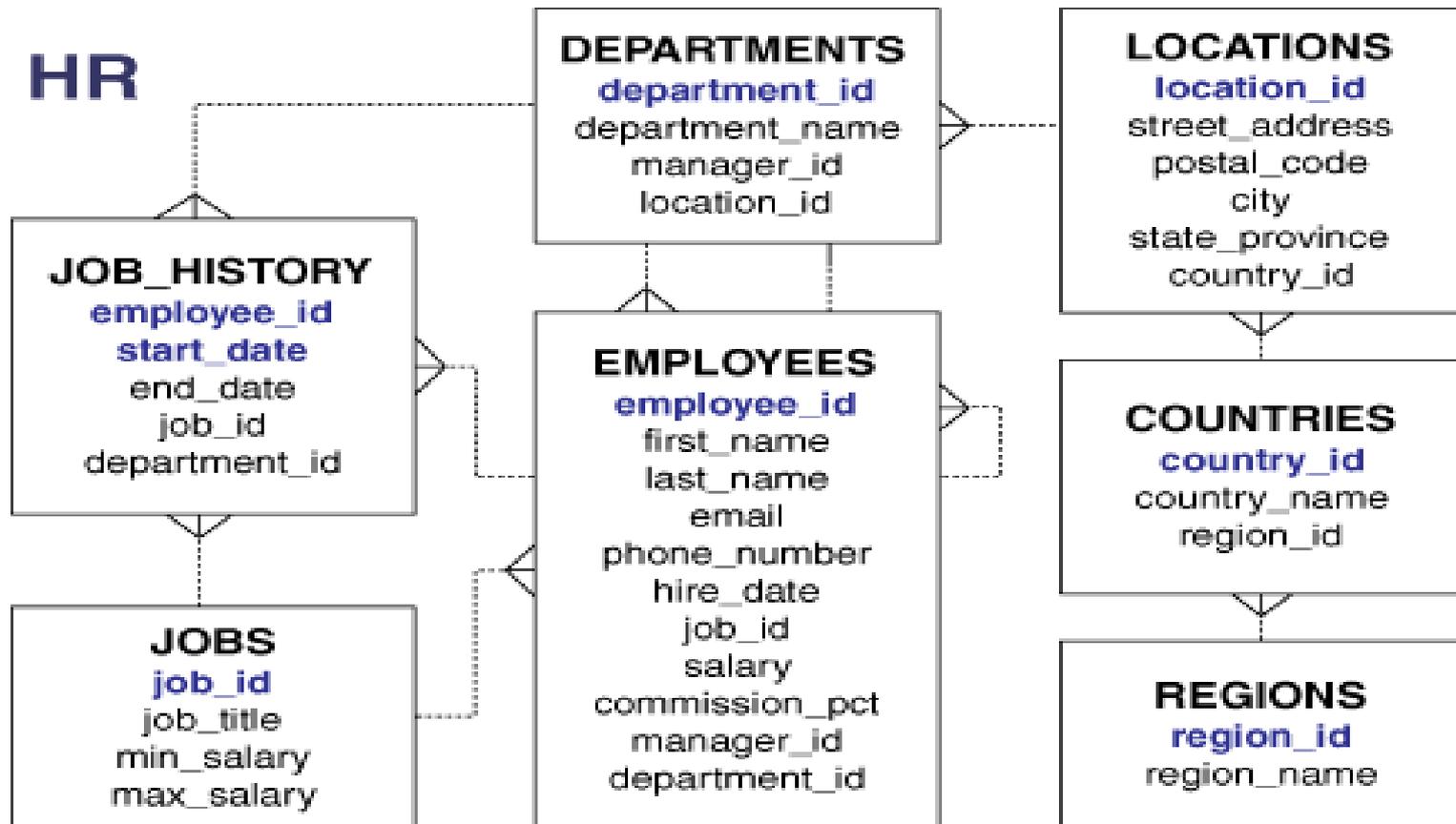
First of all will shows how to unlock the HR account and connect to Oracle Database as the user HR, who owns the HR sample schema that we use as example and tutorials in this lecture.

```
ALTER USER HR ACCOUNT UNLOCK IDENTIFIED  
BY password;
```

To display HR tables

```
SELECT * FROM tab;
```

DRL-Data Retrieval Language



HR - Entity Relation Diagram(ERD)

DRL-Data Retrieval Language

Formatting Columns in SQL-Plus

If the data in SQL-Plus displayed in multi lines and not sorted you can use Formatting Columns.

COL[UMN] {Column | Alias } [OPTIONS]

Examples:

```
COLUMN employee_id FORMAT 999
```

```
COLUMN last_name FORMAT A15
```

```
COLUMN salary FORMAT $999.99
```

```
COLUMN salary CLEAR
```

DRL-Data Retrieval Language

Syntax:

```
SELECT [DISTINCT] { * | {specific column} [[AS]
c_alias]
[, {specific column} [[AS] c_alias] ] ... }
FROM [schema.] {table|view } [t_alias]
    [, [schema.] {table|view } [t_alias] ] ...
[WHERE condition]
```

```
[GROUP BY expr [, expr] ...]
```

```
[ORDER BY {expr|position| c_alias }
[ASC|DESC] [ NULLS FIRST | NULLS LAST ]
    [, {expr|position| c_alias }
[ASC|DESC] [ NULLS FIRST | NULLS LAST ]...]
```

DRL-Data Retrieval Language

GROUP BY clause

A GROUP BY clause, part of a *Select Expression*, groups a result into subsets that have matching values for one or more columns. In each group, no two rows have the same value for the grouping column or columns. NULLs are considered equivalent for grouping purposes.

You typically use a GROUP BY clause in conjunction with an aggregate expression.

Syntax

GROUP BY

{ *column-Name* [, *column-Name*]* }

DRL-Data Retrieval Language

GROUP BY clause

Notes:

column-Name must be a column from the current scope of the query; there can be no columns from a query block outside the current scope. For example, if a GROUP BY clause is in a subquery, it cannot refer to columns in the outer query.

Select Items in the *Select Expression* with a GROUP BY clause must contain only aggregates or grouping columns.

DRL-Data Retrieval Language

Using Aggregate Functions in Queries

Aggregate functions return a single result row based on groups of rows, rather than on single rows. Aggregate functions can appear in select lists and in ORDER BY and HAVING clauses. They are commonly used with the GROUP BY clause in a SELECT statement, where Oracle Database divides the rows of a queried table or view into groups. In a query containing a GROUP BY clause, the elements of the select list can be aggregate functions, GROUP BY expressions, constants, or expressions involving one of these. Oracle applies the aggregate functions to each group of rows and returns a single result row for each group.

DRL-Data Retrieval Language

Using Aggregate Functions in Queries

List of some aggregate functions

- **COUNT**
- **FIRST**
- **LAST**
- **MAX**
- **MEDIAN**
- **MIN**
- **SUM**

DRL-Data Retrieval Language

Examples of group by Using Aggregate Functions.

Using the GROUP BY Clause: To return the number of employees for each department in the employees table, issue the following statement:

```
SELECT department_id DEPT, COUNT(*) "NO Of Emp"  
FROM employees  
GROUP BY department_id  
ORDER BY department_id;
```

DRL-Data Retrieval Language

Examples of group by Using Aggregate Functions.

The results will as below

<i>DEPT</i>	<i>NO Of Emp</i>
10	1
20	2
30	6
40	1
50	45
60	5
70	1
80	34
90	3
100	6
110	2
	1

DRL-Data Retrieval Language

Examples of group by Using Aggregate Functions.

Using the GROUP BY Clause: To return the minimum and maximum salaries for each department in the employees table, issue the following statement:

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

DRL-Data Retrieval Language

Examples of group by Using Aggregate Functions.

The results will as below

DEPARTMENT_ID	MIN(SALARY)	MAX(SALARY)
10	4400	4400
20	6000	13000
30	2500	11000
40	6500	6500
50	2100	8200
60	4200	9000
70	10000	10000
80	6100	14000
90	17000	24000
100	6900	12008
110	8300	12008
	7000	7000

DRL-Data Retrieval Language

Examples of group by Using Aggregate Functions.

To return the minimum and maximum salaries for the clerks in each department, issue the following statement:

```
SELECT department_id, MIN(salary), MAX (salary)
FROM employees
WHERE job_id = 'PU_CLERK'
GROUP BY department_id
ORDER BY department_id;
```

DRL-Data Retrieval Language

Examples of group by Using Aggregate Functions.

You can nest aggregate functions. For example, the following example calculates the average of the maximum salaries of all the departments

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

```
AVG(MAX(SALARY))
```

```
-----
```

```
10926.3333
```





Thank you

Dhafar Sabah Yaseen