

# Database System

Lecture 18

SQL Sub Languages

**DRL Data Retrieval Language**

*DRL command*

*SELECT Statement*

*Using Operators and 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*

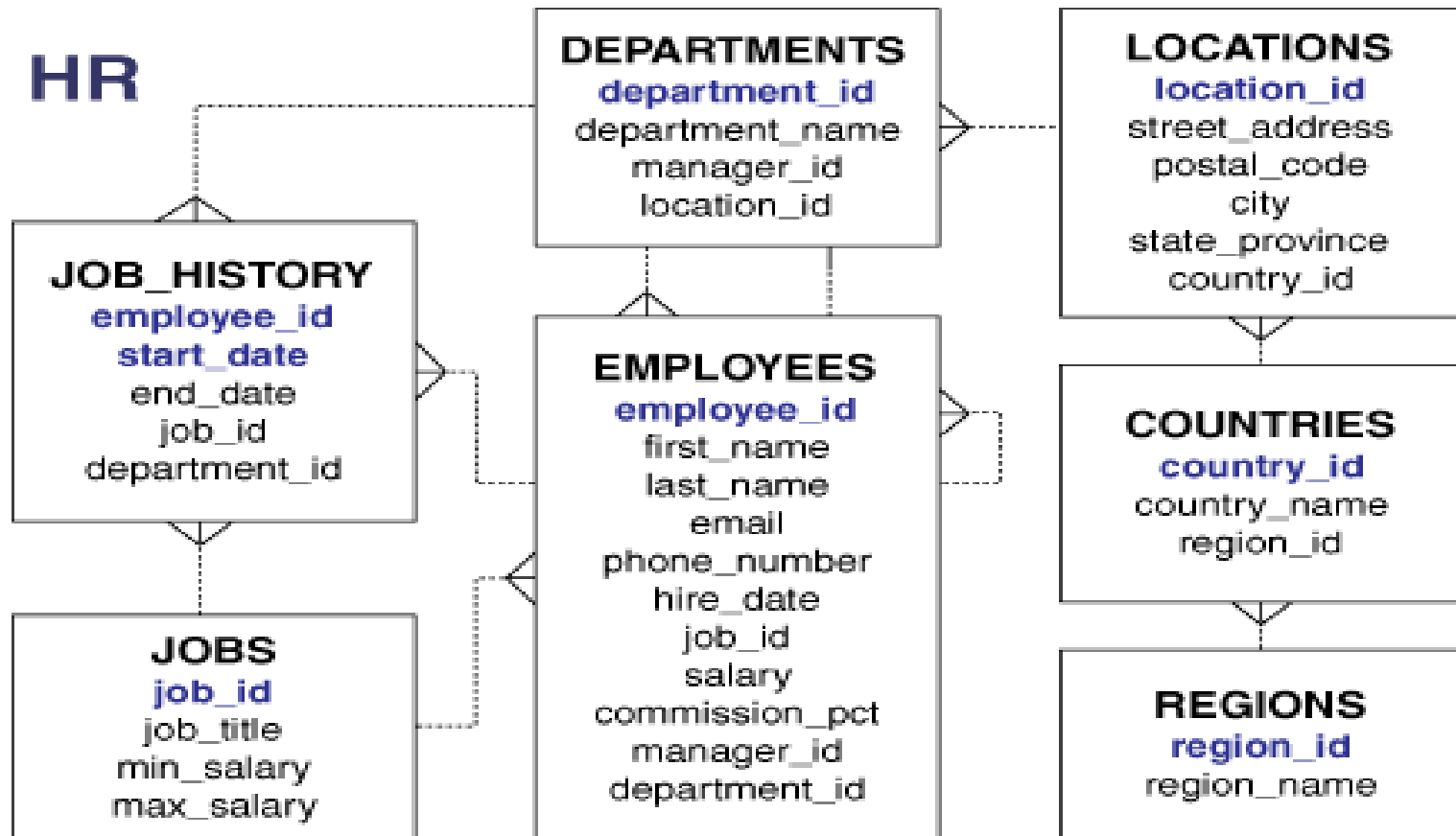
First of all will shows how to unlock the HR account and connect to Oracle Database as the user HR, who owns the HR sample schema that we use as example and tutorials in this lecture.

```
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***

## ***Using Operators and Functions in Queries***

The `select_list` of a query can include SQL expressions, which can include SQL operators and SQL functions. These operators and functions can have table data as operands and arguments. The SQL expressions are evaluated, and their values appear in the results of the query.

## ***Using Arithmetic Operators in Queries***

## ***Using Numeric Functions in Queries***

## ***Using the Concatenation Operator in Queries***

## ***Using Character Functions in Queries***

## ***Using Datetime Functions in Queries***

## ***Using Conversion Functions in Queries***



# ***DRL-Data Retrieval Language***

## ***Using Arithmetic Operators in Queries***

SQL supports the basic arithmetic operators: + (addition), - (subtraction), \* (multiplication), and / (division).

### **Example :**

```
SELECT LAST_NAME,  
       SALARY "Monthly Pay",  
       SALARY * 12 "Annual Pay"  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID = 90  
ORDER BY SALARY DESC;
```

# ***DRL-Data Retrieval Language***

## ***Using Numeric Functions in Queries***

Numeric functions accept numeric input and return numeric values. Each numeric function returns a single value for each row that is evaluated.

***Numeric Functions examples :***

***{ ABS, SIN, COS, TAN, LOG, MOD, POWER, REMAINDER, SQRT, ROUND (number), TRUNC (number) }***

# *DRL-Data Retrieval Language*

## *Example :*

```
SELECT LAST_NAME,  
ROUND (((SALARY * 12)/365), 2) "Daily Pay"  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID = 100  
ORDER BY LAST_NAME;
```

## *Another Example :*

```
SELECT LAST_NAME,  
TRUNC ((SALARY * 12)/365) "Daily Pay"  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID = 100  
ORDER BY LAST_NAME;
```

# *DRL-Data Retrieval Language*

## *Using the Concatenation Operator in Queries*

The concatenation operator (||) combines two strings into one string, by appending the second string to the first. For example, 'a' || 'b' = 'ab'. You can use this operator to combine information from two columns or expressions in the same column of the report.

### *Example:*

```
SELECT FIRST_NAME || ' ' || LAST_NAME "Name"  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID = 100  
ORDER BY LAST_NAME;
```

# *DRL-Data Retrieval Language*

## *Using Character Functions in Queries*

Character functions accept character input. Most return character values, but some return numeric values. Each character function returns a single value for each row that is evaluated.

### *Character Functions examples :*

{ LOWER, NLS\_LOWER , UPPER, NLS\_UPPER ,  
CONCAT, TRIM ,LTRIM, RTRIM, RPAD, LPAD  
,INITCAP, NLS\_INITCAP, REPLACE , ASCII,  
LENGTH }

# *DRL-Data Retrieval Language*

## *Example :*

```
SELECT UPPER(LAST_NAME) "Last",  
INITCAP(FIRST_NAME) "First",  
LOWER(EMAIL) "E-Mail"  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID = 100  
ORDER BY EMAIL;
```

# *DRL-Data Retrieval Language*

## *Using Datetime Functions in Queries*

Datetime functions operate on DATE, time stamp, and interval values. Each datetime function returns a single value for each row that is evaluated.

For each DATE and time stamp value, Oracle Database stores this information:

Year , Month , Date , Hour , Minute, Second

### *Character Functions examples :*

{ SYSDATE, EXTRACT (datetime), TO\_CHAR (datetime),  
LAST\_DAY, NEXT\_DAY, ADD\_MONTHS, CURRENT\_DATE,  
TRUNC (date), CURRENT\_TIMESTAMP , DBTIMEZONE,  
LOCALTIMESTAMP, MONTHS\_BETWEEN,  
SYSTIMESTAMP, }

# *DRL-Data Retrieval Language*

## *Example :*

```
SELECT LAST_NAME, EXTRACT(YEAR FROM SYSDATE),  
EXTRACT(YEAR FROM HIRE_DATE),  
(EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR  
FROM HIRE_DATE)) "Years Employed"  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID = 100  
ORDER BY "Years Employed";
```

The query in Example uses the EXTRACT and SYSDATE functions to show how many years each employee in department 100 has been employed. The SYSDATE function returns the current date of the system clock as a DATE value.



# *DRL-Data Retrieval Language*

## *Another Example :*

```
SELECT EXTRACT(HOUR FROM SYSTIMESTAMP)
|| ':' || EXTRACT(MINUTE FROM
SYSTIMESTAMP) || ':' ||
ROUND(EXTRACT(SECOND FROM
SYSTIMESTAMP), 0) || ', ' ||
EXTRACT(MONTH FROM SYSTIMESTAMP) || '/'
||
EXTRACT(DAY FROM SYSTIMESTAMP) || '/' ||
EXTRACT(YEAR FROM SYSTIMESTAMP) "System
Time and Date"
FROM DUAL;
```

# *DRL-Data Retrieval Language*

The table in the FROM clause of the query, DUAL, is a one-row table that Oracle Database creates automatically along with the data dictionary. Select from DUAL when you want to compute a constant expression with the SELECT statement. Because DUAL has only one row,

# *DRL-Data Retrieval Language*

## *Using Conversion Functions in Queries*

Conversion functions convert one data type to another.

### *Conversion Functions examples:*

{ TO\_CHAR (character), TO\_CHAR (datetime),  
TO\_CHAR (number), TO\_DATE, TO\_NUMBER,  
TO\_BINARY\_DOUBLE, TO\_BINARY\_FLOAT,  
TO\_BLOB, TO\_CLOB, TO\_NCHAR (character),  
TO\_NCHAR (datetime), TO\_NCHAR (number) }

# *DRL-Data Retrieval Language*

## *Example :*

```
SELECT LAST_NAME,  
HIRE_DATE,  
TO_CHAR(HIRE_DATE, 'FMMonth DD YYYY') "Date  
Started"  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID = 100  
ORDER BY LAST_NAME;
```

The query in [Example](#) uses the TO\_CHAR function to convert HIRE\_DATE values (which are of type DATE) to character values that have the format FMMonth DD YYYY. FM removes leading and trailing blanks from the month name. FMMonth DD YYYY is an example of a datetime format model.

# *DRL-Data Retrieval Language*

## *Another Example :*

```
SELECT CITY,  
POSTAL_CODE "Old Code",  
TO_NUMBER(POSTAL_CODE) + 1 "New Code"  
FROM LOCATIONS  
WHERE COUNTRY_ID = 'US'  
ORDER BY POSTAL_CODE;
```





Thank you

*Dhafar Sabah Yaseen*