Bootcamp TOC

|  |  |
| --- | --- |
| 1 | UNIX and Linux Essentials - Classroom (Campus Only) |
| 2 | XML Fundamentals - Classroom (Campus Only) |
| 3 | Oracle Database 19c: SQL workshop - Classroom (Campus Only) |
| 4 | Fundamentals of Java 17 - Classroom (Campus Only) |
| 5 | JPA with Eclipselink - Classroom (Campus Only) |
| 6 | Oracle WebLogic Server 14c: Administration I - Classroom (Campus Only) |
| 7 | REST Services & Webservices - Classroom (Campus Only) |
| 8 | Micro Services Architecture, Spring Boot features, Kafka (Basics) - Classroom (Campus Only) |
| 9 | Developing Web Applications with JavaScript, HTML5, and CSS  - Classroom (Campus Only) |
| 10 | Ojet and Angular JS, Node JS and Knockout JS - Classroom (Campus Only) |
| 11 | Introduction to Devops - Classroom (Campus Only) |

* Unix
* XML
* Oracle Database
* Java 17
* Eclipse Link for JPA
* Weblogic Server
* WebService
* Microserivce
* Web - HTML, CSS & Javasccript
* OJET
* DevOps

Case Study -> Incremental based approach

Unix

* It is a powerful OS which can be connected by multiple users at a time
* Terminal Based

:wq : Write & Quit

:q : Quits without saving

i : When you open vi editor, type “i” to insert the content

cp: to copy

mv: to rename

touch : to create files

ls : to list

vi : to edit

echo : to print

cat > : to overwrite

cat >> : to append

Search for below software in oracle-emp

1. Git -> present
2. JDK 17 -> present
3. Eclipse IDE for enterprise ->
4. Oracle Database 19c -> You must use pc name which is less than 14 characters & while installing you must use mobile hotspot
5. Weblogic Server
6. VS Code

Install Git -> Git can also be used to try all the Unix related commands, the main purpose of GIT is to collaborate | integrate everyone’s work

XML: Extensible Markup Language, it is mainly used to transform the data as well to configure the applications

Transforming: One Application can share the data in XML and another application parses the XML and converts to the structure it understands

App1 (Java) XML App2 (C#)

Application configuration: When the application want to connect to the database or want to configure server properties then they can use XML as it’s structure is strict.

ex:  
<server>  
 <port>9090</port>  
 <context-path>/api</context-path>  
 <protocol>HTTP</protocol>  
</server>

Rules of XML

1. An XML file must use tags
2. You need to have only one root tag
3. Tags must nest properly, below XML is invalid because of improper nesting

<server>  
 <port>9090 <context-path>/api</port></context-path>  
</server>

1. You can create your own XML tags & can also define/declare those tags in a schema

Two types of XML

1. Well formed that follows all the basic rules like proper nesting, case sensitivity, only one root tag
2. Valid XML: this follows Well formed + XML schema

XML Schema are of 2 types

1. DTD - Document Type Definition - Plain text syntax
2. XSD - XML Schema Definition - Uses XML syntax

XML Schema: These are the rules that declares the XML elements & their attributes, so that you can only use the tags & attributes declared in the Schema

Syntax of DTD:

<!ELEMENT element\_name (#PCDATA or child\_element) >  
<!ATTLIST element\_name attribute\_name value\_type some\_option >

PCDATA: Parsed Character Data -> You can’t use any reserved symbols here  
value\_type of an attribute can be CDATA: Character Data  
some\_option: can be #REQUIRED, #IMPLIED, #FIXED

#REQUIRED: Mandatory

#IMPLIED: Optional

#FIXED: Constants like bank\_name, college\_name,

Wild cards like: \*, +, ? would specify how many times a tag can repeat

“\*” : 0 or more  
“+” : 1 or more  
“?” : 0 or 1

XSD: XML Schema Definition, It is also used to apply schema rules, but it has wider range of types & occurrences, the rule is done using UI

XSD has many types like

* integer, long, float, string
* complex type : a tag that has child tags

RDBMS with Oracle 19C

1. What is RDBMS
2. Keys : Primary key, Composite Key, Candidate Key, Foreign Key, Super Key
3. Normalization in the database, 1NF, 2NF, 3NF
4. ER Model - Understand how to create ER diagrams

Oracle database gives us HR schema in an HR account, which has some tables which you can use to test & learn SQL

By default HR account is locked, you need to unlock.

Root user login: / as sysdba or connect / as sysdba

You need to login as a root user & alter the HR account.

Creating a user in oracle database

Note: You must be a root user, when you create a user it will not have any privilege to login or create any database resource.

Syntax:

create user username identified by password;

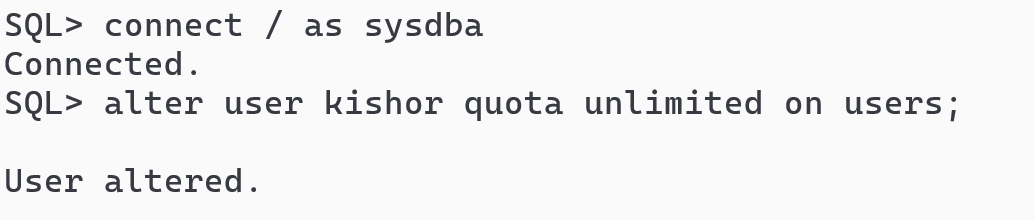
grant connect, resource to username; # this will give permission to login & create database resources

Example:

create user kishor identified by Kishor123;

In ORCLPDB, you need to set the container before you perform any task

alter session set container=orclpdb;



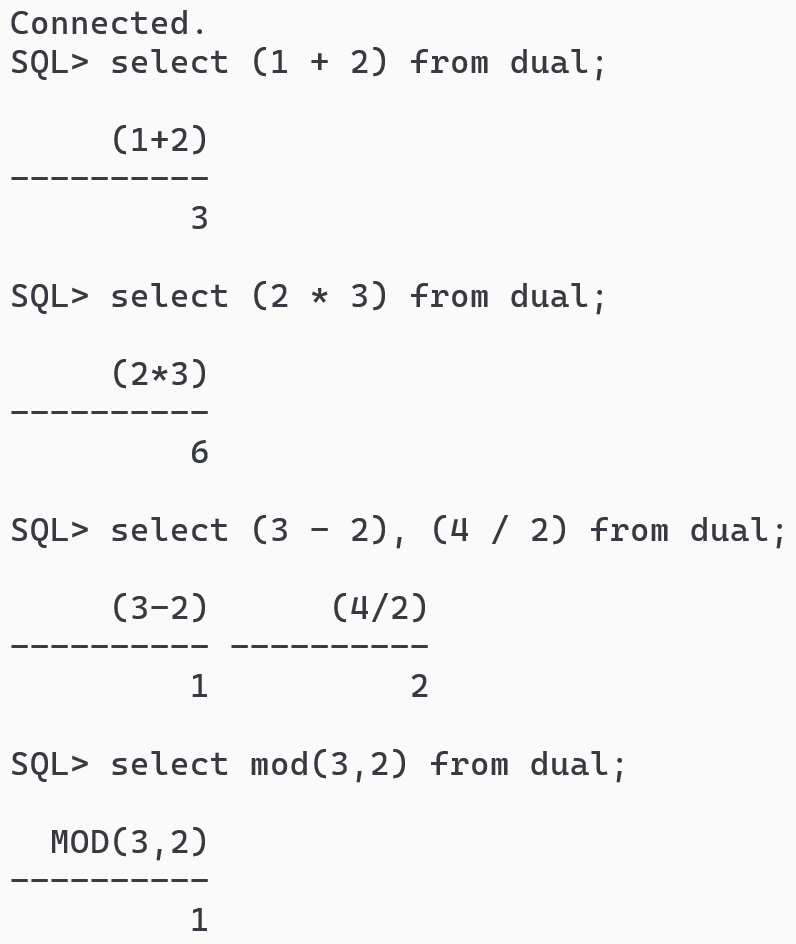
Summary:

1. Logged in as root user: / as sysdba or connect / as sysdba
2. Created the user, grant the permission to login and also set the quota
3. Unlocked the HR schema and grant the permission to login
4. We used user\_tables to identify the tables created by the logged in user

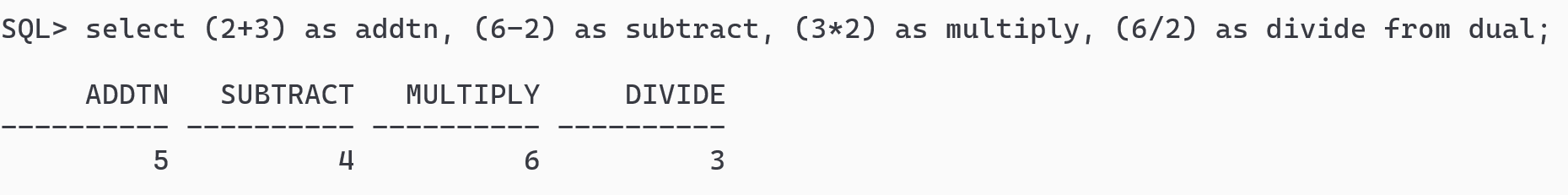
DUAL table

In oracle database it is a table used to get some guaranteed results of one row on any operation

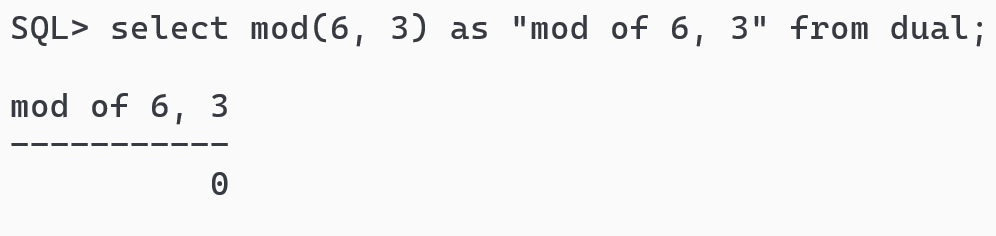
Trying some arithmetic operations using DUAL table



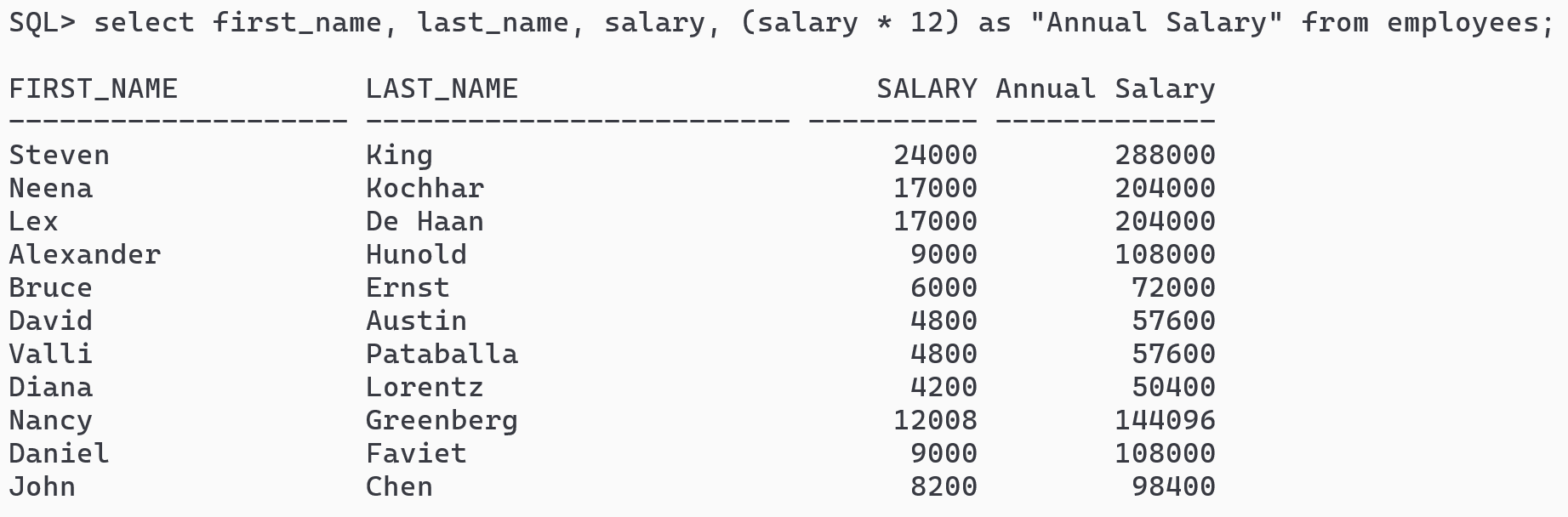
Using alias names to the columns



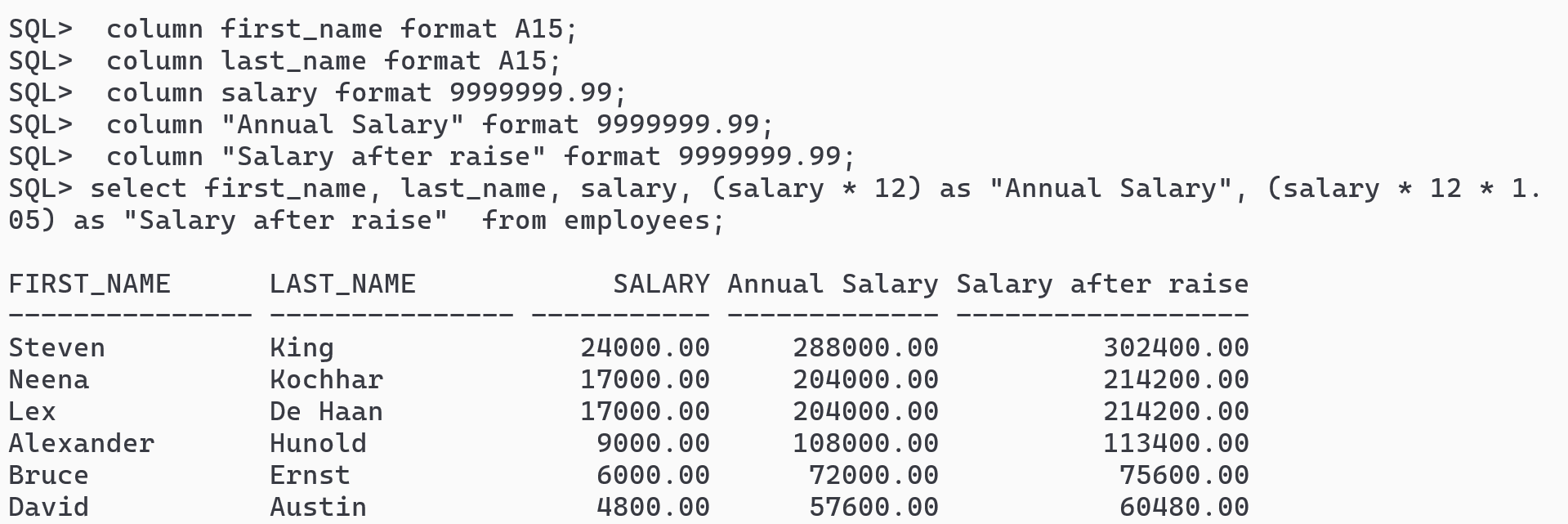
You can use double quotes to provide alias names having multiple words



Applying select & arithmetic operation on HR.EMPLOYEES table



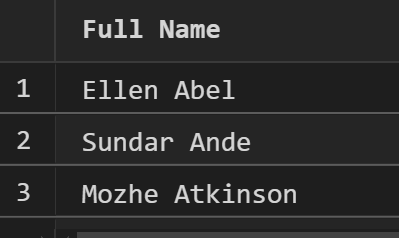
Display the result which gives incremented salary of 5% to the annual salary



Concatenation in oracle

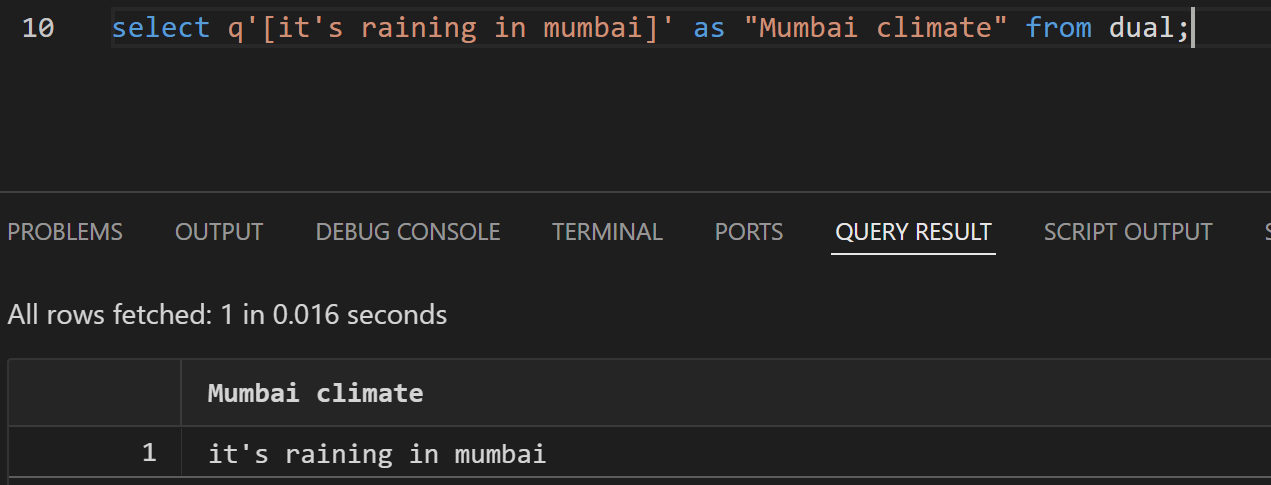


Output:



Alternative quote operator in oracle

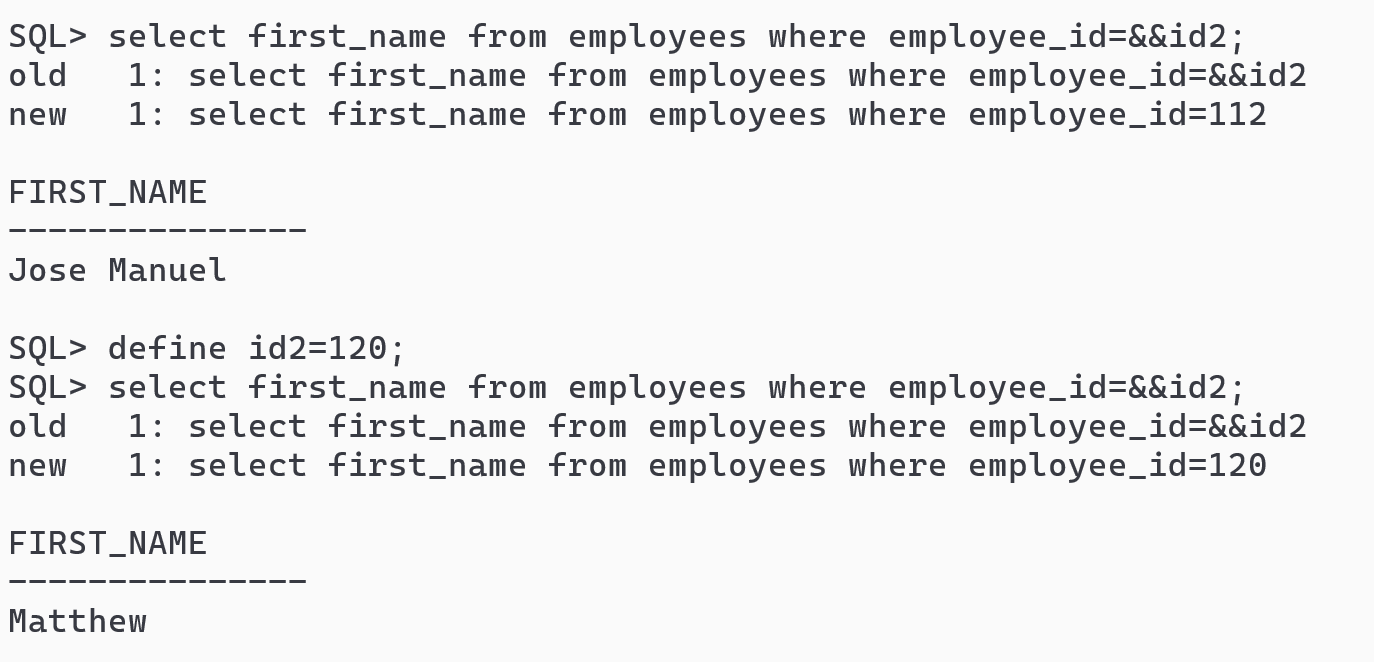
To avoid confusions while using lot of quotes in a string you can use q’[strings]’



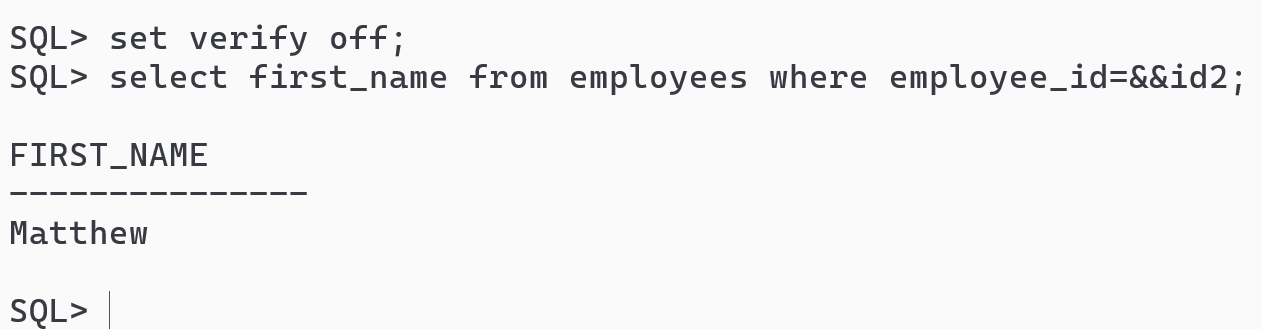
& and &&

& asks for input each time

&& asks for input only once & reuses the value, you can change the value with define.



If you wish to not see the old & new query you can use set verify off and change to on, if you wish to see the old & new query use set verify on



SQL Functions

In Oracle, **SQL functions** are built-in operations that let you **manipulate data** in queries. They're divided into two broad categories:

1. Single-Row functions: Operate on **one row at a time** and return **one result per row**.
2. Aggregate functions: Operate on **multiple rows** and return a **single result per group** or for the whole table.

Single row functions

Character functions: upper, lower, initcap, substr, replace, length, trim, concat

Numeric functions: trunc, round, floor, ceil, sqrt, abs, mod

Conversion functions: to\_char, to\_date

Date functions: sysdate, current\_date, current\_timestamp, add\_months, months\_between

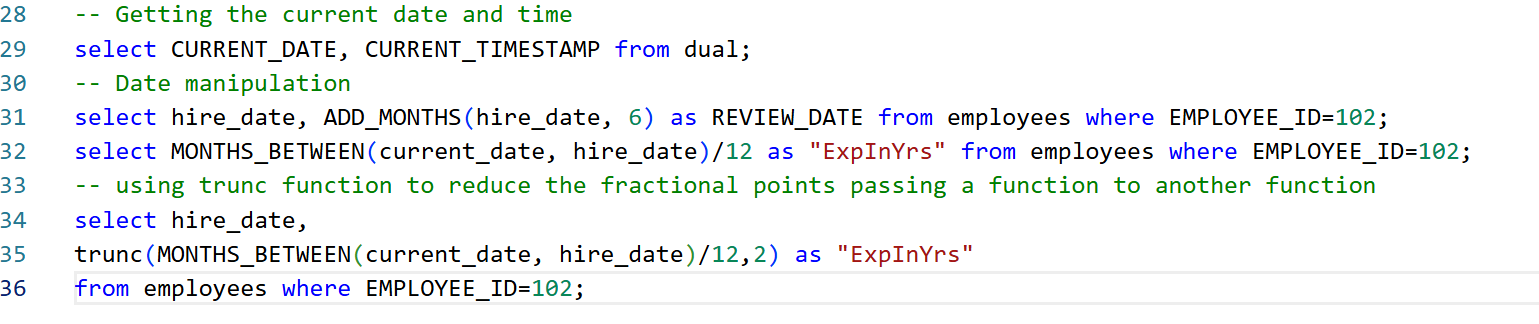
Note: sysdate, current\_date, current\_timestamp are pseudo functions, which doesn’t take parentheses

ex: select current\_date from dual;

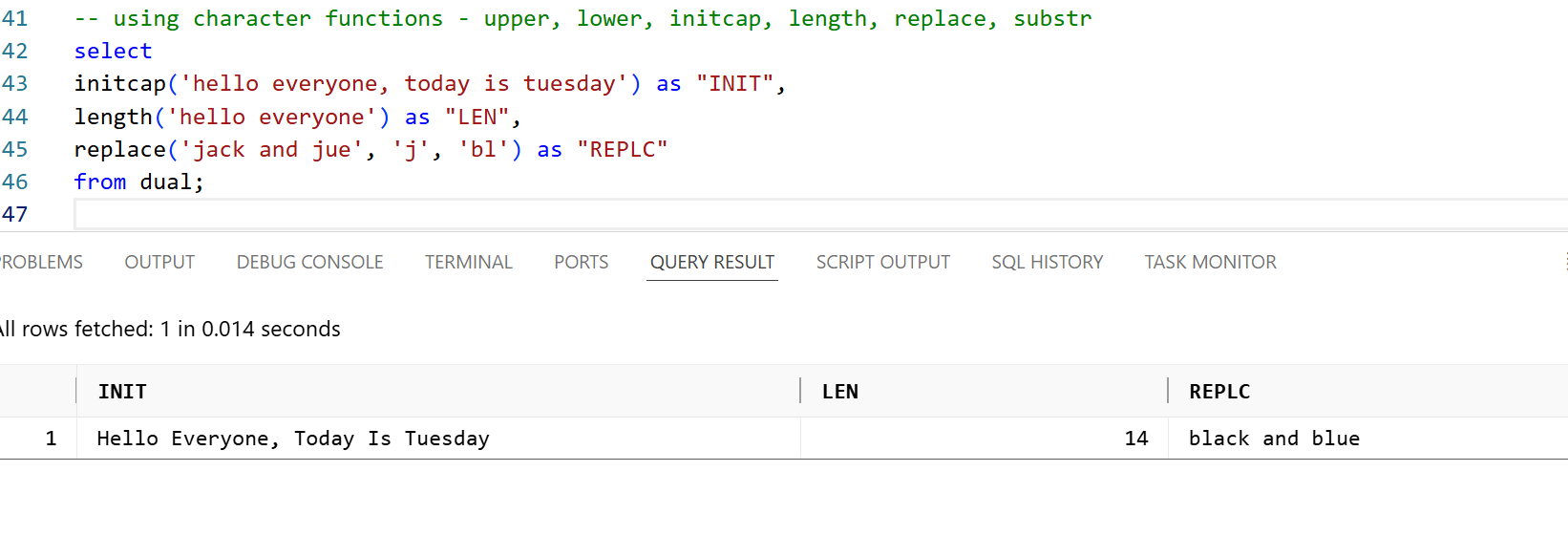
wrong: select current\_date() from dual;

sysdate & current\_date gives the same result, however sysdate uses system clock to get the date, current\_date uses the time zone of the user logged in to get the date

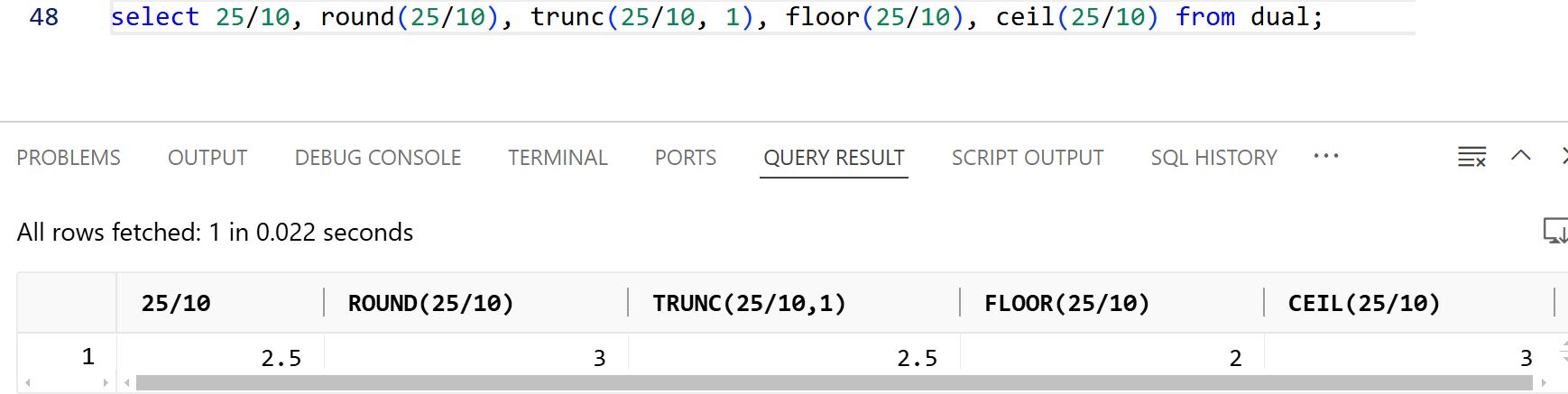
Using Date



Using Character functions



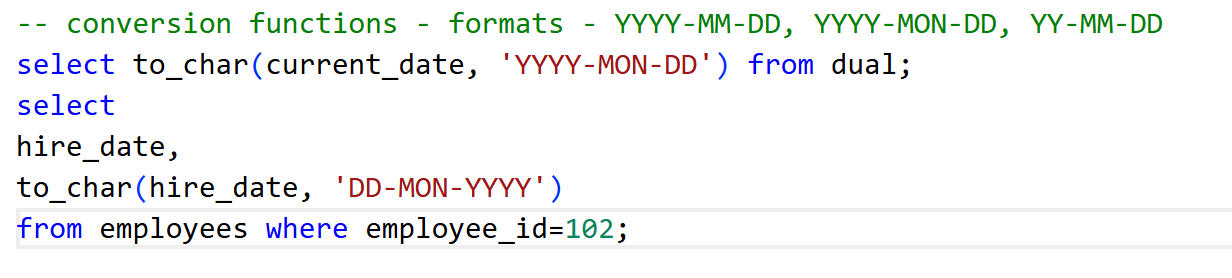
Numeric functions



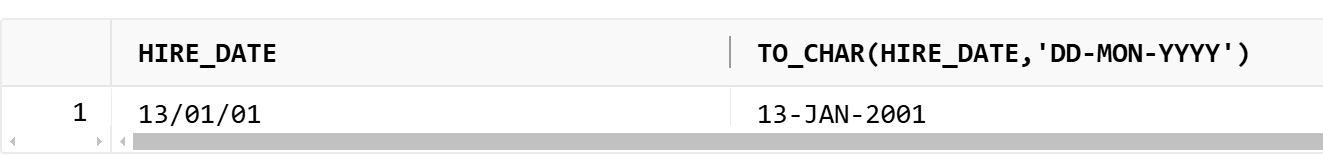
Conversion functions

to\_char: converts date to a string

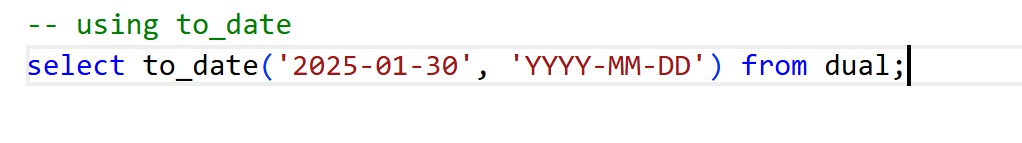
to\_date: converts string to a date



Output:



to\_date: It converts string to date



ISO-Date format

Globally ISO is the standard format which is used by most of the modern applications, the ISO format is dd-MM-yyyy, but oracle is not following ISO format

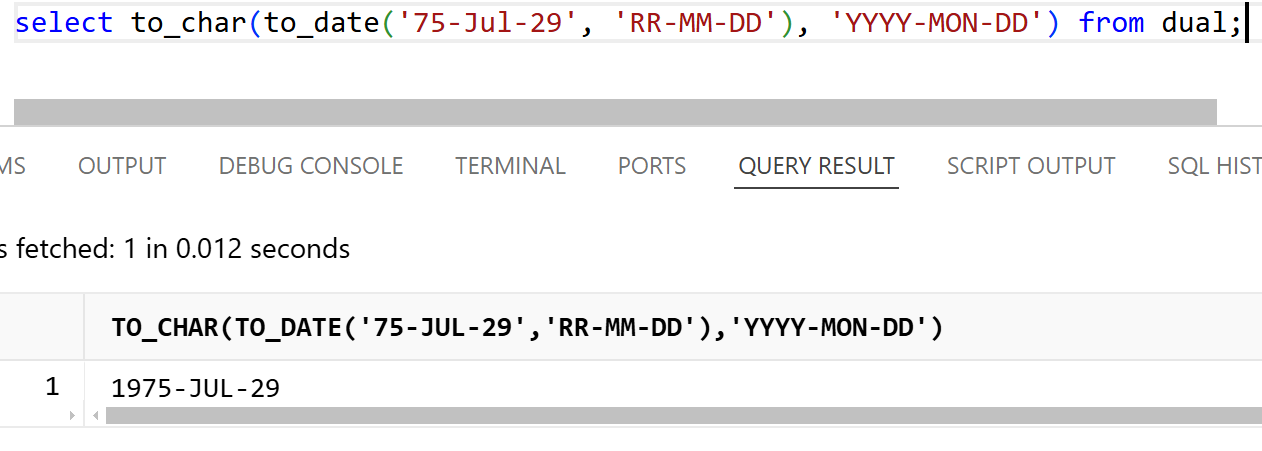
RR date format

The **RR date format** in Oracle is a **two-digit year format** that helps Oracle **guess the correct century** (1900s or 2000s) when you're working with dates.

It is especially useful when dealing with **legacy data** or **Y2K-type logic**, where users may enter only two digits for the year.

0 to 49 interpreted as 2000 to 2049

50 to 99 interpreted as 1950 to 1999

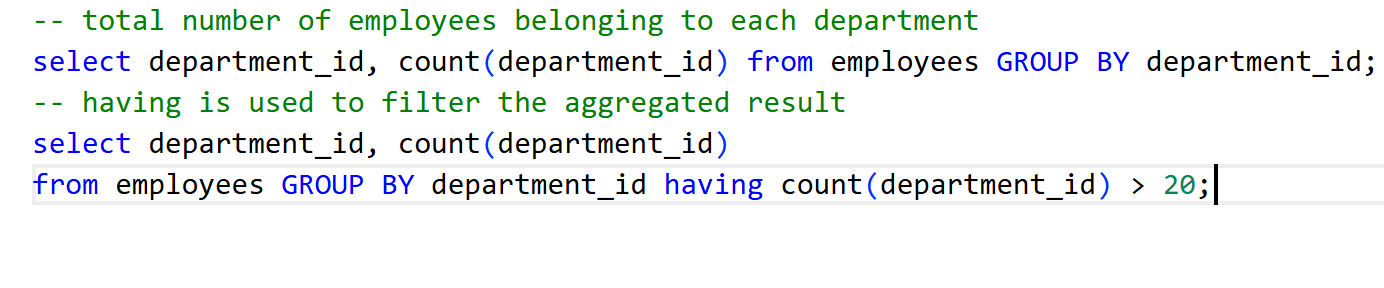


Aggregate functions

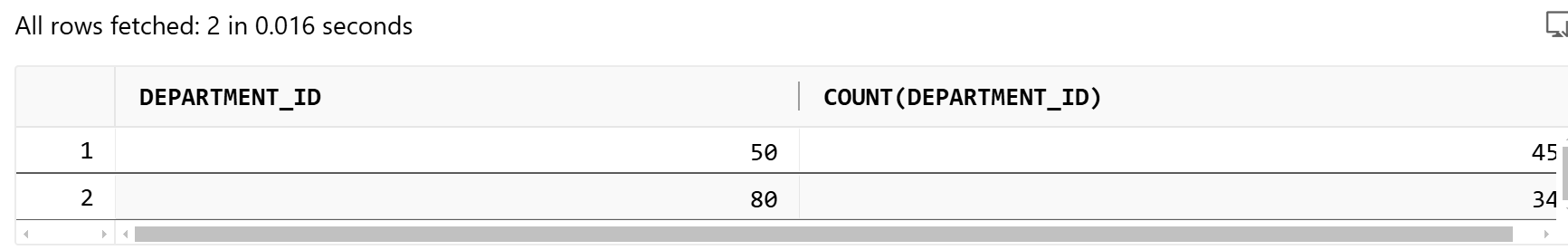
Performs operations on multiple rows & return single result

* count
* sum
* avg
* max
* min

Total number of employees belonging to each department



OUTPUT:



Activities

* Self learning till Chapter - 6
* Keep the queries in the script file.
* Create another script file that will have corresponding MYSQL queries for the queries you try in OracleDB

Evaluation at 4PM

SQL JOINS

JOINS are used to combine the rows from two or more tables based on the JOIN condition.

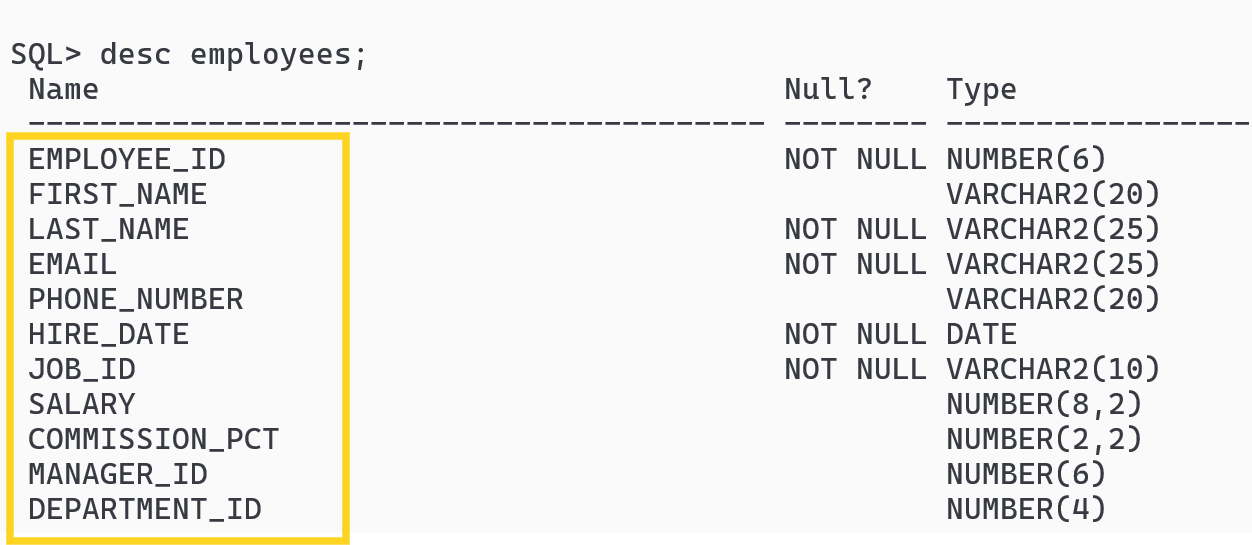
Types of JOINS

1. INNER JOIN
2. LEFT JOIN / LEFT OUTER JOIN
3. RIGHT JOIN / RIGHT OUTER JOIN
4. FULL OUTER JOIN
5. CROSS JOIN
6. SELF JOIN
7. NATURAL JOIN.

To JOIN tables we must have a clear picture about the table columns, let us find columns of below tables

1. EMPLOYEES
2. DEPARTMENTS
3. LOCATIONS
4. JOBS
5. JOBS\_HISTORY

EMPLOYEES

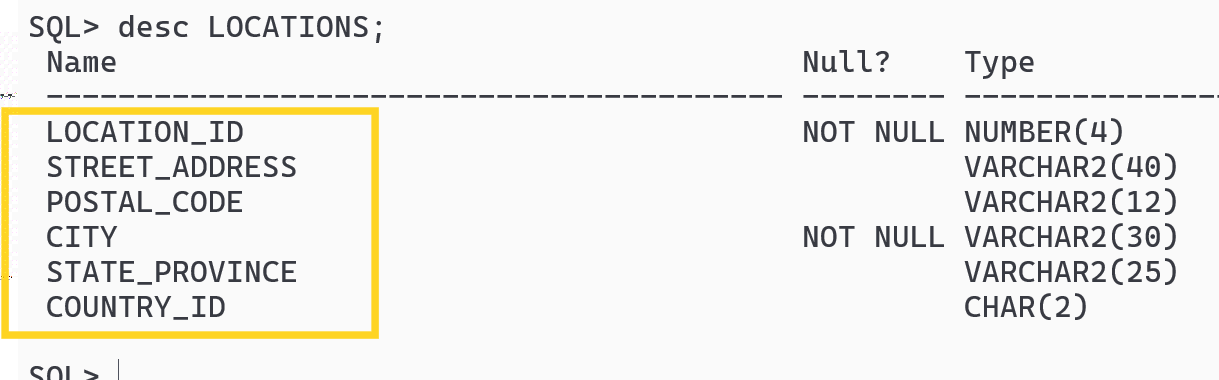


DEPARTMENTS



Common columns between EMPLOYEES & DEPARTMENTS are: DEPARTMENT\_ID, MANAGER\_ID

LOCATIONS



Common columns between DEPARTMENTS and LOCATIONS are: LOCATION\_ID

JOBS



Common columns between EMPLOYEES and JOBS are: JOB\_ID

JOBS\_HISTORY



Common columns between EMPLOYEES & JOB\_HISTORY are: EMPLOYEE\_ID, DEPARTMENT\_ID, JOB\_ID

Common columns between JOBS & JOBS\_HISTORY are: JOB\_ID

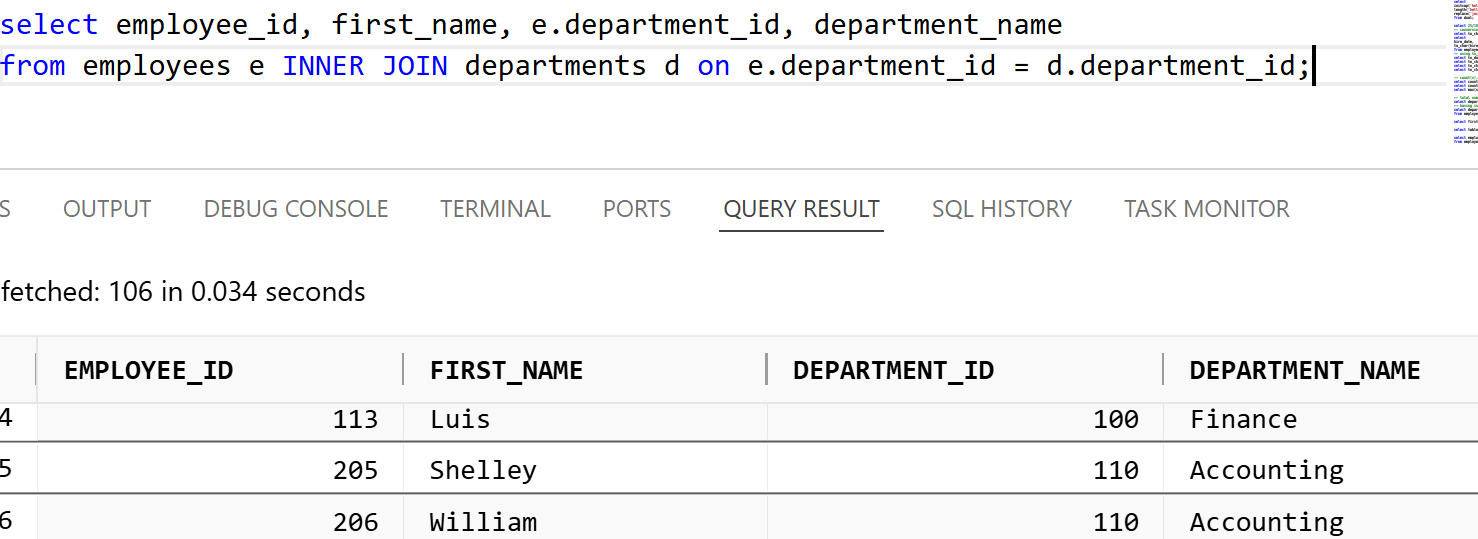
INNER JOIN

It returns only the matching rows from left and right tables based on the JOIN condition.

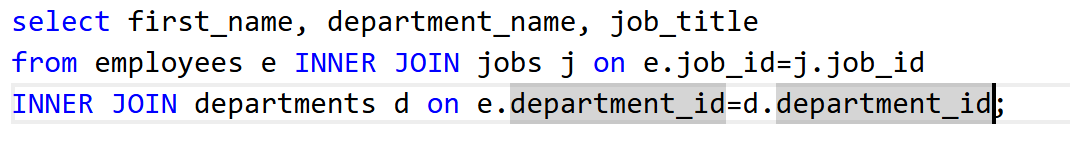
Syntax: select column1, column2 from table1 INNER JOIN table2 ON condition;

Note: There could be chance that columns with the same name in both the names, hence you need to use table aliases or full table names to select the columns, else you will get ambiguous column name error.

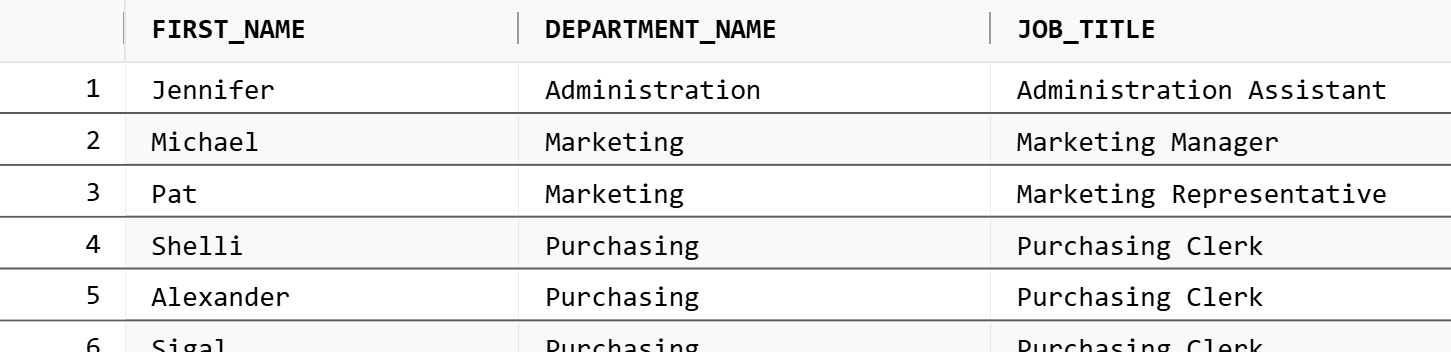
INNER JOIN on employees and departments



INNER JOIN: EMPLOYEES, JOBS and DEPARTMENTS to display each employees JOB title and their department names.

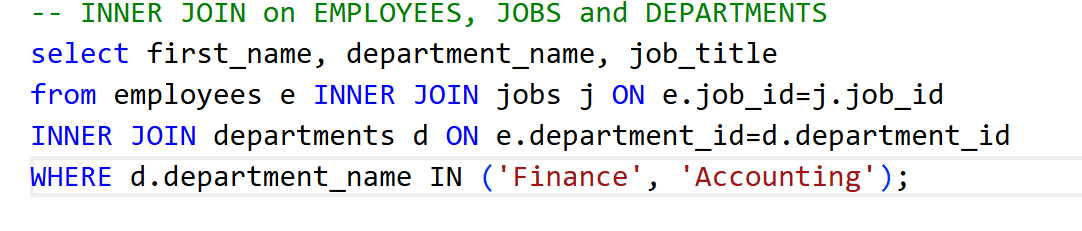


OUTPUT:

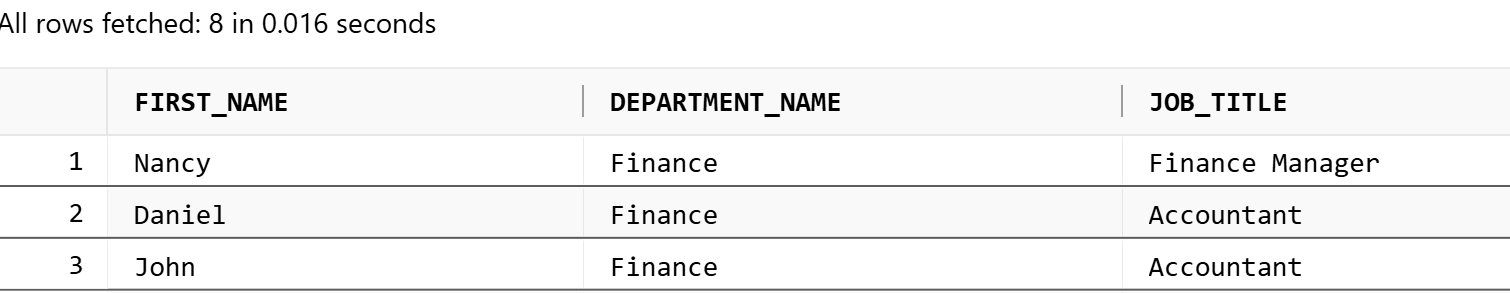


Applying filter condition using WHERE clause

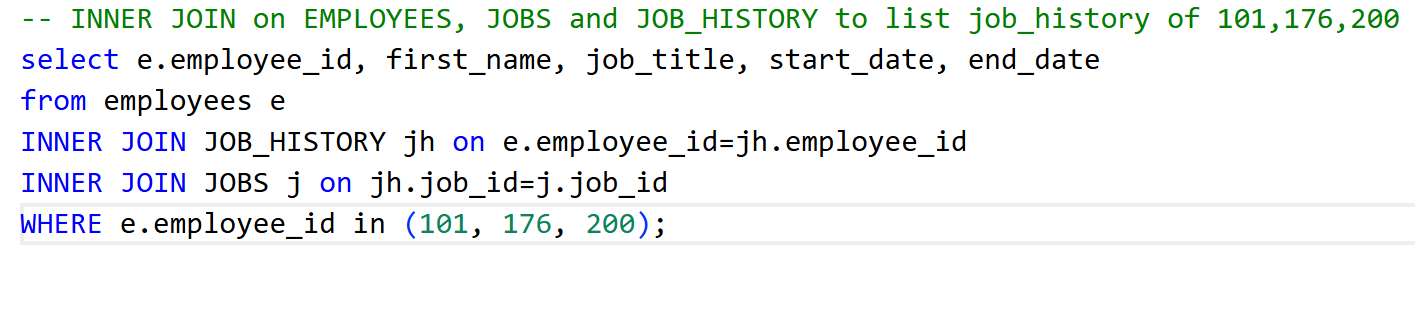
ON is used to apply condition to JOIN, WHERE is used to apply condition to filter



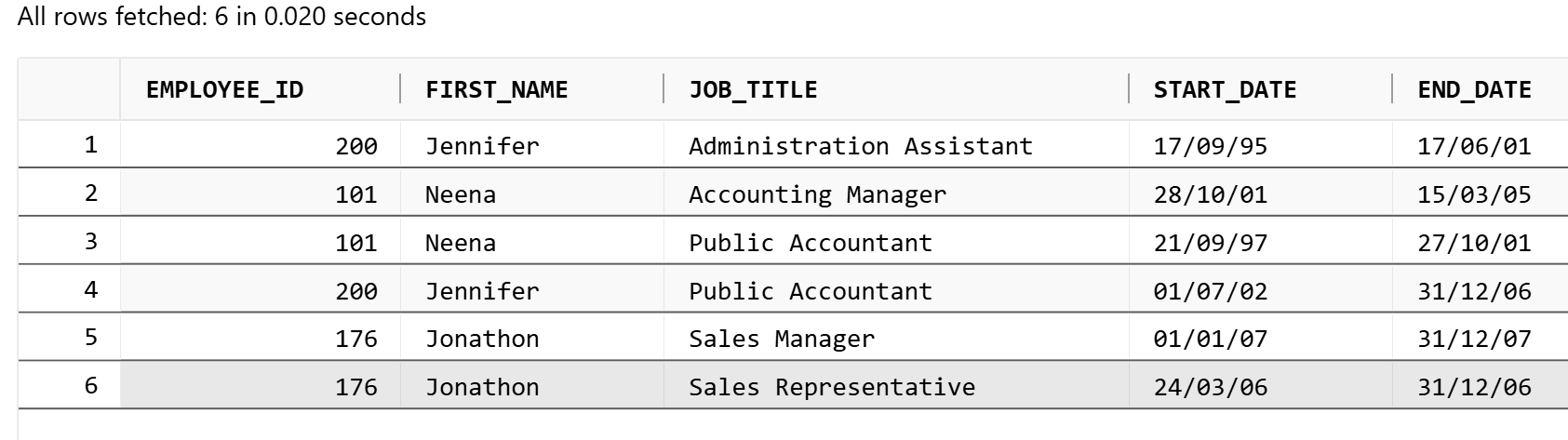
OUTPUT



Using EMPLOYEES & JOB\_HISTORY list the EMPLOYEES job\_history which will have employee\_id, first\_name, job\_title, start\_date, end\_date, filter the join that matches to the list of employee\_id: 200, 101, 176



Output



LEFT OUTER JOIN

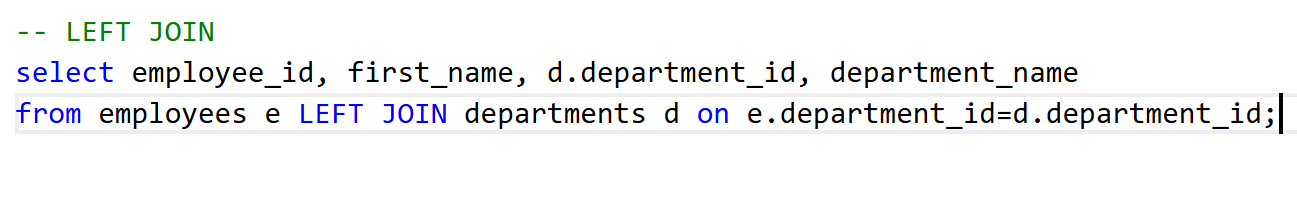
It is used to list all the rows of the LEFT table and only the matching rows of the RIGHT table.

Sometimes you want to list the employees who are not assigned to any department

syntax:

select column(s) from table1 LEFT JOIN table2 on condition

List employees & departments with LEFT JOIN based on the department\_id



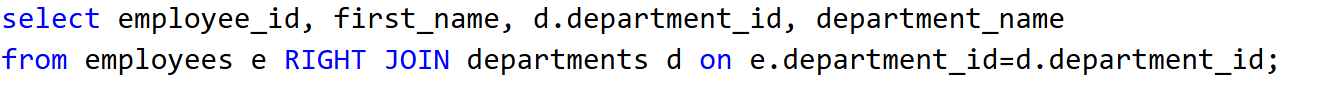
RIGHT OUTER JOIN

It is used to list all the rows of the RIGHT table and only the matching rows of the LEFT table.

Sometimes you want to list all the employees who are assigned to the department and orphan departments

syntax:

select column(s) from table1 RIGHT JOIN table2 on condition



Try the following JOINS and understand its working

1. FULL OUTER JOIN: Combines the results of left & right table
2. CROSS JOIN: Gives the cartesian product of left & right table
3. SELF JOIN: Joins the same table, this is used to list the relationship present in the same table
4. NATURAL JOIN: It uses the column names having the same name to JOIN

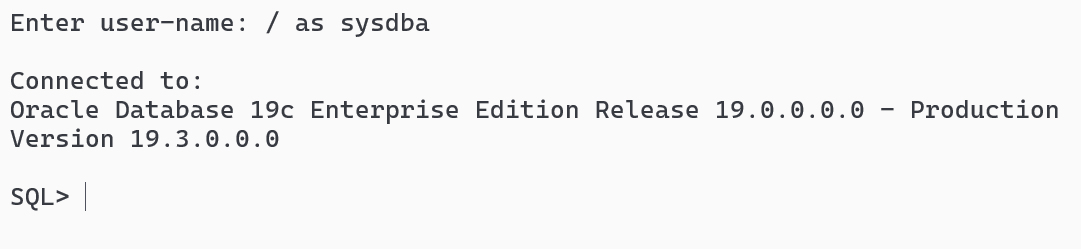
Types of SQL

1. DRL: Data Retrieval Language - SELECT, DESC
2. DDL: Data Definition Language - CREATE, ALTER, DROP, TRUNCATE
3. DML: Data Manipulation Language - INSERT, UPDATE, DELETE
4. TCL: Transaction Control Language - COMMIT, ROLLBACK
5. DCL: Data Control Language - GRANT, REVOKE

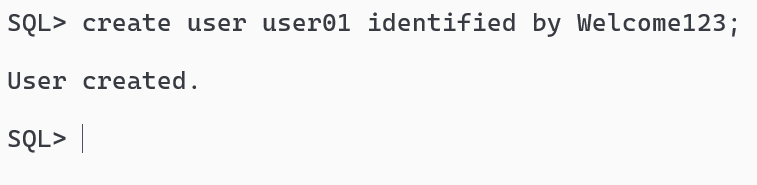
Creating user & giving connect, resource privilege

Steps to create users & give connect privilege

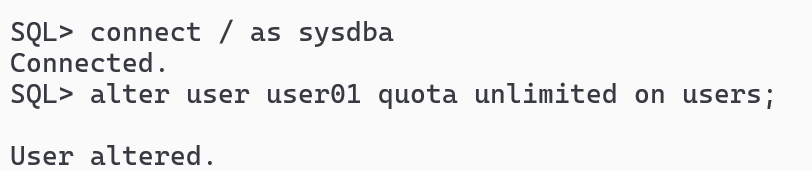
Connect / as sysdba



Create a user identified by password



Set the quota else you can’t store records

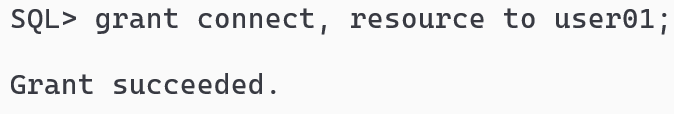


If you are using PDB then you need to first set the session

alter session set container=orclpdb;

Then you must able to create the user

Grant connect, resource permission to the user

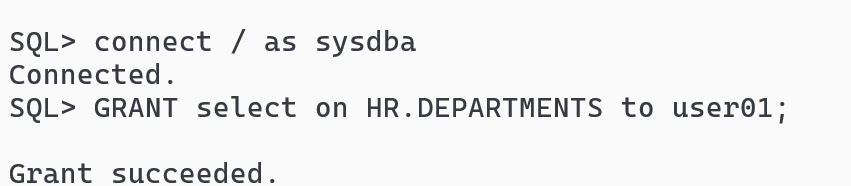


Now you must able to login as user01

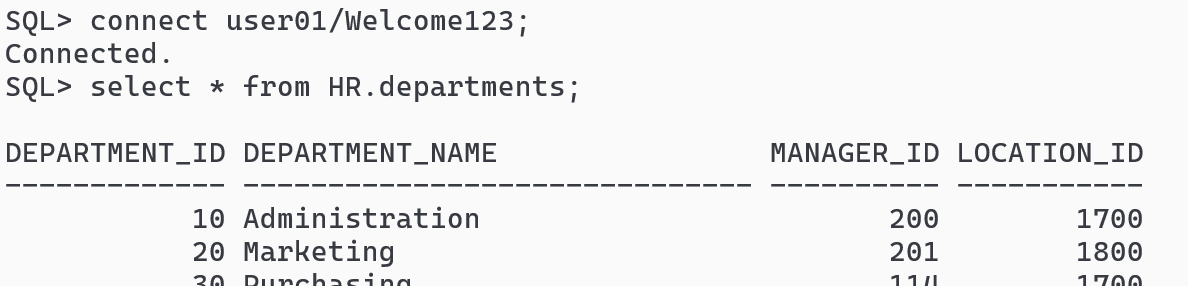


Note: The user having orclpdb must connect using user01/Welcome123@orclpdb

Can user01 access tables present in another schema



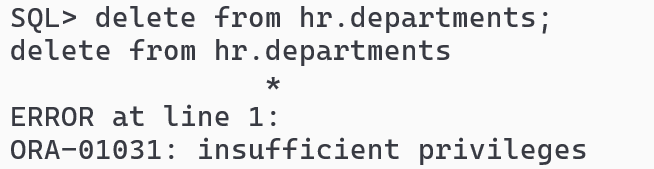
Now login as user01 and enter select command on HR.DEPARTMENTS



NOTE: If you give SELECT privilege, you can use DESC command also.

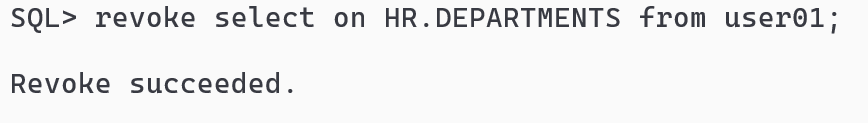
Try to access EMPLOYEES table of HR from user01, if you can’t access, give permission to user01 to read the EMPLOYEES table data.

delete from HR.departments;



How to revoke the permissions from the user

Syntax: REVOKE privilege(s) on table\_name from user\_name;



DDL

It is used to manipulate the structure of the table like create, alter, truncate & drop, you can’t rollback DDL statements i.e., you can’t undo

CREATE syntax

create table table\_name (column\_name type [constraint], ….)

Constraint: These are the rules which you can apply on a table or a column, we have the following constraints

1. PRIMARY KEY
2. UNIQUE
3. NOT NULL
4. CHECK
5. FOREIGN KEY

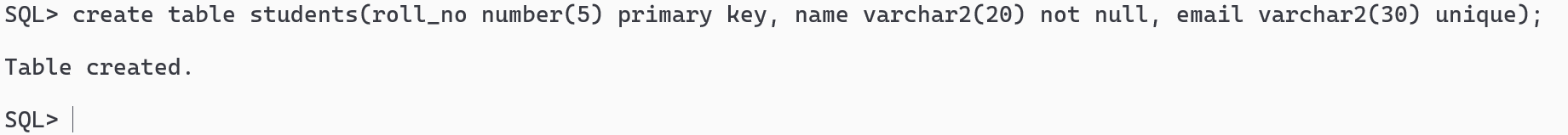
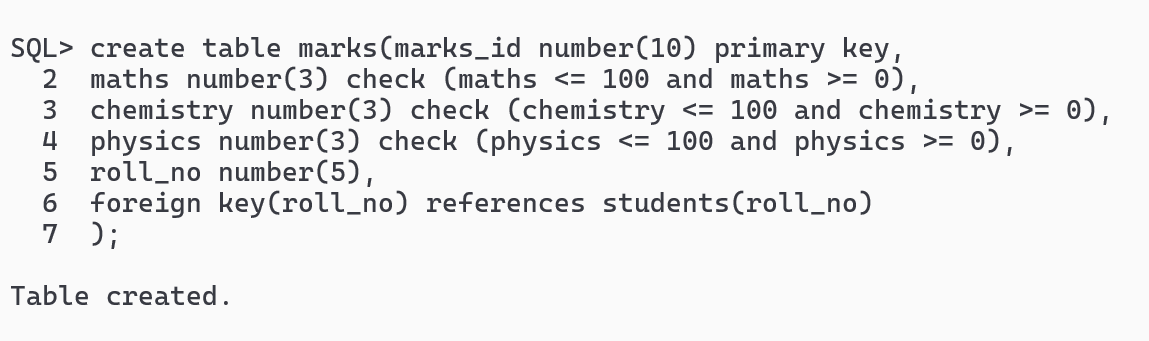


Table with CHECK & FOREIGN KEY



DML Statements

INSERT command: Store the records

Syntax:

1. insert into table\_name (col1, col2,..) values(….);
2. insert into table\_name values(….);

Display the student and marks table using join and show the total marks and their average that will have 2 fractional points (ex: 70.54)

PL/SQL

Procedural Language is a program you can store in the database and execute it whenever you need, you store SQL statements in PL/SQL, these are useful when admins want to run SQL commands everyday without rewriting the queries

There are three types of PLSQL block you can create

1. anonymous block: Doesn’t have a name, but useful to run procedures and functions
2. procedural block: Has a name
3. functional block: Similar to procedural block, but it has to mandatorily return a value

Syntax of Anonymous block

DECLARE [optional]  
 <<declaration-statement>>  
BEGIN  
 <<pl-sql statements>>  
END;

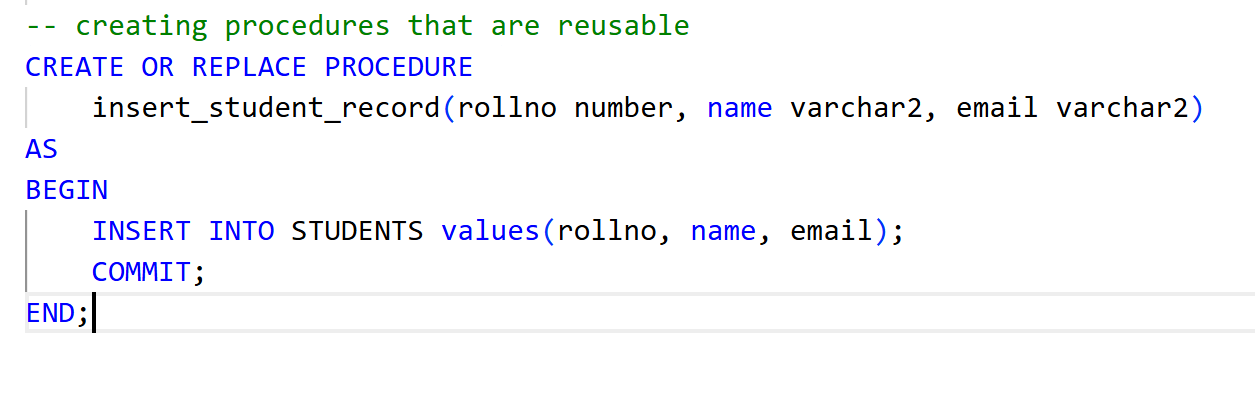
Syntax of Procedural block

CREATE OR REPLACE PROCEDURE procedure\_name(arguments)  
AS  
 <<declaration-statement>>  
BEGIN  
 <<sql statements>>  
END;

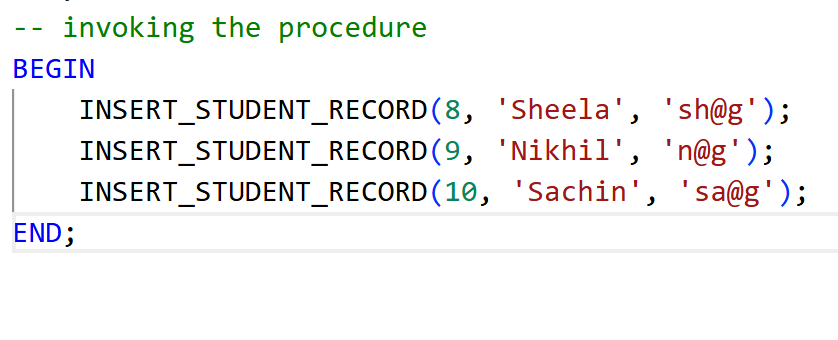
Syntax of Functional block

CREATE ORE REPLACE FUNCTION function\_name(arguments) return type  
AS  
 <<declaration-statement>>  
BEGIN  
 <<some-statements>>  
END;

Procedural block



If you must get compiled successfully to ensure procedure is created



Writing SELECT statement in PLSQL is a bit different, because you must assign the values of the columns to the PLSQL variables

SELECT column1, column2 INTO variable1, variable2 from table\_name where condition

Note: At a time you can access 1 record, if you want to access one or more records you must use FOR LOOP or CURSOR

For Self-Learning

Complete all the chapters in the TOC (Till Chapter 20)

In PLSQL write FOR LOOP to get all the student records