**\_SQL Querries\_**

| **Table Names** | **Columns** |
| --- | --- |
| Employees | Employee\_ID  First\_Name  Last \_Name  Email  Phone\_Number  Hire\_Date  Job\_ID  Salary  Commission\_PCT  Manager\_ID  Department\_ID |
| Job\_History | Employee\_ID  Start\_Date  End\_Date  Job\_ID  Department\_ID |
| Departments | Department\_ID  Department\_Name  Manager\_ID  Location\_ID |
| Countries | Country\_ID  Country\_Name  Region\_ID |
| Locations | Location\_ID  Street\_Address  Postal\_Code  City  State\_Province  Country\_ID |
| Jobs | Job\_ID  Job\_Title  Min\_Salary  Max\_Salary |
| Regions | Region\_ID  Region\_Name |

| **Questions** | **Commands** |
| --- | --- |
| **SQL SELECT** | |
| Write a query to get all the columns from the **Customers** table. | SELECT **\*** FROM Customers; |
| Write a query that will select the **City** column from the **Customers** table. | SELECT **City** FROM Customers; |
| Select **all the *UNIQUE* values** from the **Country** column in the **Customers** table. | SELECT **DISTINCT** Country FROM Customers; |
| **SQL Where** | |
| Select all records where the City column has the value "Berlin". | SELECT \* FROM Customers  WHERE City = 'Berlin'; |
| Use the NOT keyword to select all records where City is  **NOT "Berlin"**. | SELECT \* FROM Customers  WHERE **NOT** City = 'Berlin'; |
| Select all records where the CustomerID column has the value 32. | SELECT \* FROM Customers  WHERE CustomerID = 32; |
| Select all records where the City column has the value 'Berlin' and the PostalCode column has the value 12209. | SELECT \* FROM Customers  WHERE City = 'Berlin'  **AND** PostalCode = 12209; |
| Select all records where the City column has the value 'Berlin' **or** 'London'. | SELECT \* FROM Customers  WHERE City = 'Berlin'  **OR** City = 'London'; |
| **SQL Order By** | |
| Select all records from the Customers table, sort the result alphabetically by the column City. | SELECT **\*** FROM Customers  ORDER BY City; |
| Select all records from the Customers table, sort the result reversed alphabetically by the column City. | SELECT **\*** FROM Customers ORDER BY City **DESC**; |
| Select all records from the Customers table, sort the result alphabetically, first by the column Country, then, by the column City. | SELECT **\*** FROM Customers  ORDER BY Country, City; |
| **SQL Insert** | |
| Insert a new record in the Customers table. | **INSERT INTO** Customers **(**CustomerName, Address, City,  PostalCode, Country**)**  **VALUES (**'Hekkan Burger', 'Gate Veien 15', 'Sandnes', '4306', 'Norway'**)**; |
| **SQL Null** | |
| Select all records from the Customers where the PostalCode column is empty. | SELECT **\*** FROM Customers  WHERE PostalCode **IS NULL**; |
| Select all records from the Customers where the PostalCode column is NOT empty. | SELECT **\*** FROM Customers  WHERE PostalCode **IS NOT NULL**; |
| SQL Update | |
| Update the City column of all records in the Customers table. | **UPDATE** Customers  **SET** City = 'Oslo'; |
| Set the value of the City columns to 'Oslo', but only the ones where the Country column has the value "Norway". | **UPDATE** Customers  **SET** City = 'Oslo'  Where Country = 'Norway'; |
| Update the City value and the Country value for CustomerID =32 | **UPDATE** Customers  **SET** City = 'Oslo'**,** Country = 'Norway'  WHERE CustomerID = 32; |
| **SQL Delete** | |
| Delete all the records from the Customers table where the Country value is 'Norway'. | DELETE FROM Customers  Where Country = 'Norway'; |
| Delete all the records from the Customers table. | DELETE FROM Customers; |
| **SQL Functions** | |
| Use the MIN function to select the record with the **smallest** value of the Price column. | SELECT **MIN**(Price)  FROM Products; |
| Use an SQL function to select the record with the **highest** value of the Price column. | SELECT **MAX**(Price)  FROM Products; |
| Use the correct function to return the **number of records** that have the Price value set to 18. | SELECT **COUNT**(\*)  FROM Products  WHERE Price = 18; |
| Use an SQL function to calculate the average price of all products. | SELECT **AVG**(Price)  FROM Products; |
| Use an SQL function to calculate the sum of all the Price column values in the Products table. | SELECT **SUM**(Price)  FROM Products; |
| **SQL Like** | |
| Select all records where the value of the City column starts with the letter "a". | SELECT \* FROM Customers  WHERE City **LIKE** '**a%**'; |
| Select all records where the value of the City column ends with the letter "a". | SELECT \* FROM Customers  WHERE City **LIKE** '**%a**'; |
| Select all records where the value of the City column contains the letter "a". | SELECT \* FROM Customers  WHERE City **LIKE** '**%a%**'; |
| Select all records where the value of the City column starts with letter "a" and ends with the letter "b". | SELECT \* FROM Customers  WHERE City **LIKE** '**a%b**'; |
| Select all records where the value of the City column does NOT start with the letter "a". | SELECT \* FROM Customers  WHERE City **NOT LIKE** '**a%**'; |
| **SQL Wildcards** | |
| Select all records where the second letter of the City is an "a". | SELECT \* FROM Customers  WHERE City **LIKE** '**\_a%**'; |
| Select all records where the first letter of the City is an "a" or a "c" or an "s". | SELECT \* FROM Customers  WHERE City **LIKE** '**[acs]%**'; |
| Select all records where the first letter of the City starts with anything from an "a" to an "f". | SELECT \* FROM Customers  WHERE City **LIKE** '**[a-f]%**'; |
| Select all records where the first letter of the City is NOT an "a" or a "c" or an "f". | SELECT \* FROM Customers  WHERE City **LIKE** '**[!acf]%**'; |
| **SQL In** | |
| Use the IN operator to select all the records where Country is either "Norway" or "France". | SELECT \* FROM Customers  Where Country **IN** ('Norway', 'France'); |
| Use the IN operator to select all the records where Country is NOT "Norway" and NOT "France". | SELECT \* FROM Customers  Where Country **NOT** **IN** ('Norway', 'France'); |
| **SQL Between** | |
| Use the BETWEEN operator to select all the records where the value of the Price column is between 10 and 20. | SELECT \* FROM Products  WHERE Price **BETWEEN** 10 **AND** 20; |
| Use the BETWEEN operator to select all the records where the value of the Price column is NOT between 10 and 20. | SELECT \* FROM Products  WHERE Price **NOT** **BETWEEN** 10 **AND** 20; |
| Use the BETWEEN operator to select all the records where the value of the ProductName column is alphabetically between 'Geitost' and 'Pavlova'. | SELECT \* FROM Products  WHERE ProductName **BETWEEN** 'Geitost' **AND** 'Pavlova'; |
| **SQL AS** | |
| When displaying the Customers table, make an ALIAS of the PostalCode column, the column should be called Pno instead. | SELECT CustomerName, Address, PostalCode **AS** Pno  FROM Customers; |
| When displaying the Customers table, refer to the table as Consumers instead of Customers. | SELECT \* FROM Customers **AS** Consumers; |
| **SQL Join** | |
| Insert the missing parts in the JOIN clause to join the two tables Orders and Customers, using the CustomerID field in both tables as the relationship between the two tables. | SELECT \* FROM Orders **LEFT JOIN** Customers  **ON** Orders.CustomerID **=** Customers.CustomerID; |
| Choose the correct JOIN clause to select all records from the two tables where there is a match in both tables. | SELECT \* FROM Orders **INNER JOIN** Customers  **ON** Orders.CustomerID **=** Customers.CustomerID; |
| Choose the correct JOIN clause to select all the records from the Customers table plus all the matches in the Orders table. | SELECT \* FROM Orders **RIGHT JOIN** Customers  **ON** Orders.CustomerID **=** Customers.CustomerID; |
| **SQL Group** | |
| List the number of customers in each country. | SELECT COUNT(CustomerID), Country FROM Customers  **GROUP BY** Country; |
| List the number of customers in each country, ordered by the country with the most customers first. | SELECT COUNT(CustomerID), Country FROM Customers  **GROUP BY** Country  **ORDER BY** COUNT(CustomerID) Desc; |
| **SQL Database** | |
| Write the correct SQL statement to create a new database called testDB. | **CREATE DATABASE** testDB; |
| Write the correct SQL statement to delete a database named testDB. | **DROP DATABASE** testDB; |
| Write the correct SQL statement to create a new table called Persons. | **CREATE TABLE** Persons (  PersonID int,  LastName varchar(255),  FirstName varchar(255),  Address varchar(255),  City varchar(255)  ); |
| Write the correct SQL statement to delete a table called Persons. | **DROP TABLE** Persons; |
| Use the TRUNCATE statement to delete all data inside a table. | **TRUNCATE TABLE** Persons; |
| Add a column of type DATE called Birthday. | **ALTER TABLE** Persons  **ADD** Birthday DATE; |
| Delete the column Birthday from the Persons table. | **ALTER TABLE** Persons  **DROP COLUMN** Birthday; |
| [Oracle Link](http://local.school.portnov.com:4517/apex/f?p=4550:1:2831225135203071::::FSP_AFTER_LOGIN_URL:%5Cf?p=4500%7C1003%7C6057098607330310%7C%7CNO%7C%7C%7C%5C) **workspace: hr 383 user: HR password: hr** | |
| Company cost for employees salaries | Select SUM(Salary) from Employees |
| Average salary in IT department | Select AVG(Salary) from Employees  Where Department\_ID = 60 |
| Least compensated employee | Select MIN(Salary) from Employees |
| Find out employees who started to work for the company between 2001 and 2006  *77 rows* | SELECT \* FROM Employees  WHERE Hire\_DATE BETWEEN '1/1/2001' AND '12/31/2006'; |
| Find out how many sales representatives are in the company  *30* | SELECT COUNT(Job\_ID) FROM Employees  WHERE Job\_ID = 'SA\_REP'; |
| List all salaries in IT department  *9000*  *4800*  *4200*  *6000* | SELECT DISTINCT Salary FROM Employees  WHERE Job\_ID = 'IT\_PROG'; |
| Who is an employee with ID 199? 1009?  *Douglas Grant (Employee\_ID:199)* | SELECT \* FROM Employees  WHERE EMPLOYEE\_ID IN (199, 1009); |
| List all countries defined in the database table  *25 rows* | SELECT Country\_Name FROM Countries; |
| Find out who is manager of Corporate Tax Dept (several queries are needed; do not try to accomplish in one SQL statement)  *No data found* | SELECT \* FROM Employees  WHERE Department\_ID='130'; |
| What is the max salary Sales Manager can get?  *$20,080* | SELECT MAX\_Salary  FROM Jobs  WHERE Job\_Title = 'Sales Manager'; |
| Find out who is working on 0.25 commission  *David Bernstein*  *Peter Hall*  *Sarath Sewall*  *Clara Vishney*  *Lisa Ozer*  *Alyssa Hutton* | SELECT First\_Name, Last\_Name FROM Employees  WHERE COMMISSION\_PCT IN (0.25); |
| How many employees report to manager with ID 145?  *6* | SELECT COUNT (\*) FROM Employees  WHERE Manager\_id IN (145); |
| Get list of people names who report to manager with ID 103  *Bruce Ernst*  *David Austin*  *Valli Pataballa*  *Diana Lorentz* | SELECT First\_Name, Last\_Name  FROM Employees  WHERE Manager\_ID = 103; |
| Find out how much Luis Popp is making a month  *6900* | SELECT Salary FROM Employees  WHERE First\_Name = 'Luis' and Last\_Name = 'Popp' |
| Find Kee Gee phone number  *No data* | SELECT Phone\_Number FROM Employees  WHERE First\_Name = 'Kee' and Last\_Name = 'Gee' |
| Find first and last names of user RMATOS email address  *RANDALL MATOS* | SELECT First\_Name, Last\_Name  FROM Employees  WHERE Email = 'RMATOS' |
| Find people who work on commission basis  *35 Rows* | SELECT \* FROM Employees  WHERE Commission\_Pct >= 0 |
| \*\*\* Employees with no commission  *72 Rows* | SELECT \* FROM Employees  WHERE Commission\_Pct IS NULL  OR Commission\_Pct = 0 |
| Find phone number of employee #123  *650.123.4234* | SELECT phone\_number FROM employees  WHERE employee\_Id = 123 |
| Retrieve complete user information whose phone number is 011.44.1644.429263  *178 Kimberly Grant KGRANT ... ... ... ...* | SELECT \* FROM employees  WHERE Phone\_Number = '011.44.1644.429263'; |
| Front desk clerk needs to know if there is an employee with a name Donald O’Connell  *No data* | SELECT \* FROM employees  WHERE First\_Name = 'Donald' and  Last\_Name = 'O''Connell';  Or  SELECT \* FROM Employees WHERE First\_Name ='Donald' AND Last\_Name = q'[O'Connell]';  Or  SELECT \* FROM Employees WHERE First\_Name ='Donald' AND Last\_Name = 'O"Connell'; |
| How long employee Renske Ladwig work for the company  *18.4 years* | SELECT Round((SYSDATE - Hire\_Date)/365.25, 1)  As "Service at Company"  FROM Employees  WHERE First\_Name = 'Renske' and  Last\_Name = 'Ladwig'; |
| Advanced: find out WHERE physically HR department is located (country, city)  *London, UK* | SELECT location\_id FROM departments  WHERE DEPARTMENT\_NAME = 'Human Resources'  SELECT city, country\_id FROM locations  WHERE location\_id = 2400  Or Nested:  SELECT city, country\_id FROM locations  WHERE location\_id = (SELECT location\_id FROM departments WHERE DEPARTMENT\_NAME = 'Human Resources') |
| Find out what is salary range for Shipping clerk position  *2500 - 5500* | SELECT Min\_Salary|| '-'|| Max\_Salary  As "Range Salary" FROM jobs  WHERE Job\_Title = 'Shipping Clerk' |
| Find out how many employees work outside of US  *38* | SELECT location\_id FROM locations WHERE COUNTRY\_ID <> 'US'  SELECT department\_id FROM departments WHERE location\_id IN (SELECT location\_id FROM locations WHERE COUNTRY\_ID <> 'US')  Or Nested:  SELECT Count(Employee\_ID) FROM Employees  WHERE Department\_ID IN  (SELECT Department\_ID FROM Departments  WHERE Location\_ID IN  (SELECT Location\_ID FROM locations  WHERE Country\_ID <> 'US')) |
| Find out what department employees work FROM outside US work in?  *Marketing*  *Human Resources*  *Public Relations*  *Sales* | SELECT Department\_Name FROM Departments  WHERE Location\_ID IN  (SELECT Location\_ID FROM locations  WHERE Country\_ID <> 'US') |
| Find out how many employees work in Bay Area  *45* | make another table and input all zip codes in the bay area and make a query comparing in the employees' table  = get a map and specify the geo points of borderlines for the bay area  = SPATIAL data Microsoft SQL server try it  = Location table, look at the address  Other option (but limited)  Select Count(\*) from Employees  Where Phone\_Number Like '650%' or  Phone\_Number Like '510%' or  Phone\_Number Like '408%' or  Phone\_Number Like '415%' |
| Find out employees who do not report to the manager  *Steven King* | SELECT \* FROM Employees  WHERE Manager\_ID is Null |
| Find out who is getting the highest commission  *John Russell* | SELECT First\_Name, Last\_Name FROM Employees  WHERE Commission\_PCT = (SELECT Max(Commission\_PCT) FROM Employees) |
| Find out who is Employee #1 (hired first)  *Lex De Haan* | SELECT First\_Name, Last\_Name FROM Employees  WHERE Hire\_Date = (SELECT Min(Hire\_Date) FROM Employees) |
| Advanced: find out how many people work in each department | SELECT COUNT(EMPLOYEES.EMPLOYEE\_ID), DEPARTMENTS.DEPARTMENT\_NAME FROM EMPLOYEES JOIN DEPARTMENTS ON EMPLOYEES.DEPARTMENT\_ID=DEPARTMENTS.DEPARTMENT\_ID GROUP BY DEPARTMENTS.DEPARTMENT\_NAME |
| How many people work in each department | Select Department\_ID, Count(\*)  From Employees  Group By Department\_ID |
| Find out who (First name, Last Name) gets the top salary in each Department |  |
| Count 6 and more, Count aggregate doesn't work with WHERE (predicate) instead use **HAVING** | Select Department\_ID, Count(Employee\_ID) from Employees  Having Count(Employee\_ID) >5  Group by Department\_ID |
| In [statistics,](https://docs.google.com/spreadsheets/d/1rKva_g2gBWglRqLnCJj2ue5tSXnKD3O7PV6cgwx74L0/edit?usp=sharing) percentiles are used to understand and interpret data. The *n*th percentile of a set of data is the value at which *n* percent of the data is below it. What you need to find out is the salary percentile for employee with the name David Austin, who is currently making 4800. You need to find out the salary range for his position and then calculate (in SQL) his percentile. If it turns out that he earns min salary for the position then the percentile is 0, if max salary - 100%.  *13.33 th percentile* | *https://careertrend.com/how-4604073-calculate-salary-range-percentiles.html*  Select Round(((4800-Min\_Salary)/(Max\_Salary-Min\_Salary)\*100),2)  as "Percentile" from Jobs  Where Job\_ID IN (Select Job\_ID from Employees  Where First\_Name = 'David' and Last\_Name = 'Austin') |
| Calculate ratio of avg salaries in IT dept and Sales dept.  *0.64* | Select  Round((Select AVG(Salary) from Employees  Where Department\_ID IN (Select Department\_ID From Departments  Where Department\_Name = 'IT'))  /  (Select AVG(Salary) from Employees  where Department\_ID IN (Select Department\_ID From Departments  Where Department\_Name = 'Sales')),2) as "Ratio of Average Employees"  From Dual |
| Find out number of employees in all departments except IT and Administration.  *101*  *Total number of employees - IT and Admin*  Or *NOT IN (‘IT’, ‘Sales’) OR Is NULL* | Select  (Select Count(Employee\_ID) as "Number of Employees"  from Employees)  -  (Select Count(Employee\_ID) as "IT/Sales" from Employees  where Department\_ID IN (Select Department\_ID from Departments  Where Department\_Name IN ('IT','Administration')))  from dual |
| Company plans to hire Public Relations Representative and pay medium salary for that position. Find out that number.  *7500* | Select (Max\_Salary + Min\_Salary)/2  as "Average Salary" from Jobs  Where Job\_Title = 'Public Relations Representative'; |
| Where department NOC is located?  *2004 Charade Rd, Seattle, Washington, 98199*  *US* | Select Street\_Address, City, State\_Province, Postal\_Code, Country\_ID from Locations  Where Location\_ID IN (Select Location\_ID From Departments  Where Department\_Name = 'NOC'); |
| How many countries are in region 1?  *8* | Select Count(Country\_Name) from Countries  Where Region\_ID = 1; |
| Find out year with biggest number of hired employees  *2005* | Select \* From  (SELECT EXTRACT(year FROM Hire\_Date) "Year",  COUNT(Hire\_date) "No. of Hires"  FROM Employees  GROUP BY EXTRACT (year FROM Hire\_Date)  ORDER BY "No. of Hires" DESC)  where rownum=1; |
| Insert 1 row  \*\*\* REMOVE ✅ in Autocommit, should be BLANK  So you can undo the changes | INSERT Into Employees (Employee\_ID, First\_Name, Last\_Name)  Values (1047, 'Rea', 'Tabofunda') |
| Insert 2 rows  \*\*\* REMOVE ✅ in Autocommit, should be BLANK  So you can undo the changes | INSERT Into Employees (Employee\_ID, First\_Name, Last\_Name)  Values ({1048, 'Rea', 'Tabofunda'},  {1049, 'Carl', 'Tabofunda'}) |
| Insert sequencer for primary key  \*\*\* REMOVE ✅ in Autocommit, should be BLANK  So you can undo the changes | INSERT Into Employees Values ((Select Max(Employees\_ID)+1 from Employees), 'Rea', 'Tabofunda', 'RTAB', '408.123.0000', '01-May-85', 'IT\_PROG', 120000, null, 103, 60); |
| Permanently | Commit |
| how do you add records to a table? | INSERT into table\_name VALUES (‘ALEX’, 33 , ‘M’); |
| **Undo changes if Autocommit is unchecked**  **Or restores deleted records** | Rollback |
| Deleting entire table | DELETE from Employees |
| Deleting records for Employee 207  \*\*\* USE Primary Key in deleting records  \*\*\* REMOVE ✅ in Autocommit, should be BLANK  So you can undo the changes | DELETE from Employees  Where Employee\_ID = 207 |
| Deleting multiple employees (send email to investors & unemployment Agency about layoff)  \*\*\* REMOVE ✅ in Autocommit, should be BLANK  So you can undo the changes | DELETE from Employees  Where Emlpoyee\_ID Between 107 and 127 |
| Truncate = delete table contents irreversibly, cannot UNDO, faster than Delete |  |
| Remove the entire employee table  DROP = deletes table from database, cannot UNDO | Drop table Employees |
| Update Employee 106’s salary to 20000 | Update Employees Set Salary = 20000  Where Employee\_ID =106 |
| How do you add a column to a table? | ALTER TABLE Department ADD (AGE, NUMBER); |
| How do you change the value of the field? | UPDATE EMP\_table set number = 200 where item\_munber = ‘CD’;  update name\_table set status = 'enable' where phone = '4161112222';  update SERVICE\_table set REQUEST\_DATE = to\_date ('2006-03-04 09:29', 'yyyy-mm-dd hh24:MM') where phone = '4161112222'; |
| Add 3% to all employees’ salary | Update Employees Set Salary = Salary\*1.03 |
| Employee Diana Lorentz got married and changed her last name. Her new name is Diana Goetz | Update Employees set  Last\_Name = 'Goetz',  EMAIL = ‘DGOETZ’  Where Employee\_ID = 107 |
| **JOINS** | |
| First and Last Name of Employees, Phone Numbers and Department Name  *106 only missed 1 employee NULL department* | Select e.First\_Name, e.Last\_Name, e.Phone\_Number, d.Department\_Name  from Employees e left Join Departments d  on e.Department\_ID = d.Department\_ID; |
| First, Last name , dept ID of employees that are abroad | select e.first\_name, e.last\_name, e.department\_id from  employees e join departments d  on e.department\_id = d.department\_id  join locations l  on d.location\_id = l.location\_id  where l.country\_ID <> 'US' |
| First and Last Name of Employees, Phone Numbers and Department Name  *107 employees* | Explicitly JOINED & Table Alias  Select  e.First\_Name,  e.Last\_Name,  e.Phone\_Number,  d.Department\_Name  From Employees e LEFT OUTER JOIN Departments d  ON e.Department\_ID = d.Department\_ID |
| Retrieves info for all departments even without employees | Select  e.First\_Name,  e.Last\_Name,  e.Phone\_Number,  d.Department\_Name  From Employees e RIGHT OUTER JOIN Departments d  ON e.Department\_ID = d.Department\_ID |
| Write a select Statement which will retrieve employees first, last names, and their office address | Select  e.First\_Name "First Name",  e.Last\_Name "Last Name",  l.Street\_Address || ', ' ||  l.City || ', ' ||  l.State\_Province || ', ' ||  l.country\_ID || ', ' ||  l.postal\_code "Address"  from employees e  join departments d  on e.department\_id = d.department\_id  join locations l on l.location\_id = d.location\_id |
| Write a select statement which will print employees first and last names, salary and range of salary for their position | SELECT  E.First\_Name "First Name",  E.Last\_Name "Last Name",  E.Salary "Salary",  J.Min\_Salary || ' - ' ||  J.Max\_Salary "Salary Range"  FROM Employees E  JOIN Jobs J  ON E.Job\_ID = J.JOB\_ID |
| Write a select statement which will print following information: Department name, department Manager FIrst and Last Name | Select  d.Department\_Name,  d.Manager\_ID,  e.First\_Name,  e.Last\_Name  from Employees e JOIN Departments d  ON d.Manager\_ID = e.Manager\_ID |
| Write a select statement which will print the following information: Department Name, Department Address | Select  d.Department\_Name,  l.Street\_Address || ', ' ||  l.City || ', ' ||  l.State\_Province || ', ' ||  l.country\_ID || ', ' ||  l.postal\_code "Address"  from departments D  join Locations L  on d.location\_id = l.location\_id |
| How much is the top salary which employee #101 can earn?  *30000* | Select J.Max\_Salary from Jobs J join Employees E  on J.Job\_ID = E.Job\_ID  where E.Employee\_ID = 101 |
| Office street address where employee RMATOS (email) is working  *2011 Interiors Blvd* | Select l.Street\_Address from employees e  join departments d on e.department\_id = d.department\_id  join locations l on l.location\_id = d.location\_id  where e.email = 'RMATOS' |
| Where IT department is located (country, city, address)  *2014 Jabberwocky Rd, Southlake, Texas, US* | Select  l.Street\_Address || ', ' ||  l.City || ', ' ||  l.State\_Province || ', ' ||  l.country\_ID "Address"  from Locations l  join Departments d on d.location\_id = l.location\_id  Where Department\_Name = 'IT' |
| How much money employees working in GB are making combined?  *311000* | Select SUM(e.Salary) "Total GB Salary"  from Employees e join departments d  on e.Department\_ID = d.Department\_ID  join Locations l  On d.Location\_ID = l.Location\_ID  Where l.Country\_ID ='UK' |
| List all employees and their POSSIBLE max salary from Jobs table | Select e.First\_Name, e.Last\_Name, j.Max\_Salary  from Employees e join Jobs j  on e.Job\_ID = j.Job\_ID |
| GROUPING |  |
| HOw many people work in each department | Select Department\_ID, Count(\*) from EMployees  Group by Department\_ID |
| View  Virtual Table | New\_SAPR\_Employees = Select \* from Employees  where job\_ID = 'SA\_Rep' and Hire\_Date > '01/01/2007'  Select \* from (Select \* from Employees  where job\_ID = 'SA\_Rep' and Hire\_Date > '01/01/2007')  Select \* From New\_SAPR\_Employees |
| Self Join  Count how many employees report to each manager | Select e2.Employee\_ID, Count(e2.Employee\_ID)  From Employees e1, Employees e2  Where e1.Manager\_ID = e2.Employee\_ID  Group by e2.Employee\_ID |
| Select Random collection of rows on the table  *10% of the rows* | Select \* From Employees  Sample(10) |
|  | Select Distinct  Unique |
| Hire\_date in Employees table represents employee's hire date. Naturally, start\_date in Job\_history table cannot be smaller than hire\_date. Write a SQL query which will check this condition for all listings in job\_history table. | Select e.Employee\_ID, e.Hire\_date "Hiring Date",  j.Start\_date "Start Day"  from Employees e Join Job\_History j  ON e.Employee\_ID = j.Employee\_ID  Where j.Start\_date < e.Hire\_date |
| Calculate salary percentile for all employees | Select e.First\_Name ||' '|| e.Last\_Name "Employee Name",  Round(((e.Salary- j.Min\_Salary)/(j.Max\_Salary-j.Min\_Salary)\*100),2)  as "Percentile Salary"  from Employees e Join Jobs j  ON e.Job\_ID = j.Job\_ID |

**PRACTICE PROBLEMS**

| **PRACTICE Questions**  Locations, Departments, Jobs & Employees tables | **Commands** |
| --- | --- |
| List all the employees’ details  *107 rows* | Select \* from Employees; |
| List all the department details  *27 rows* | Select \* from Departments; |
| List all jobs details and order by the Max-Salary.  *19 rows* | Select \* from Jobs; |
| List all the locations order by the city in alphabetical order.  *23 rows* | Select \* from Locations  Order By City; |
| List only the fields first name,last name,salary, commission for all employees  *107 rows* | Select First\_Name, Last\_Name, Salary, Commission\_PCT from Employees; |
| List out employee\_id,last name,department id for all employees and rename employee id as “ID of the employee”, last name as “Name of the employee”, department id as “department ID”  *107 rows* | Select Employee\_ID as "ID of the employee", Last\_Name as "Name of the employee", Department\_ID as "department ID"  from Employees; |
| List out the employees’ annual salary with their names only  *107 rows* | Select First\_Name, Last\_Name, Salary  from Employees; |
| List the details about “SMITH”  *Lindsey Smith*  *William Smith* | Select \* from Employees  Where Last\_Name = 'Smith' or  First\_Name ='Smith'; |
| List out the employees who are working in department 20  *Michael Hartstein, Pat Fay* | Select First\_Name, Last\_Name From Employees  Where Department\_ID = 20; |
| List out the employees who are earning salary between 3000 and 4500  *23 rows* | Select First\_Name, Last\_Name From Employees  Where Salary BETWEEN 3000 AND 4500; |
| List out the employees who are working in department 10 or 20  *Jennifer Whalen,* *Michael Hartstein, Pat Fay* | Select First\_Name, Last\_Name from Employees  Where Department\_ID IN (10,20); |
| Find out the employees who are not working in department 10 or 30  *99 rows* | Select First\_Name, Last\_Name from Employees  Where Department\_ID NOT IN (10,30); |
| List out the employees whose Last name starts with “S”  *9 rows* | Select Last\_Name from Employees  where Last\_Name like 'S%'; |
| List out the employees whose Last name start with “S” and end with “N”  *Sullivan* | Select Last\_Name from Employees  where Last\_Name like 'S%n'; |
| List out the employees whose Last name length is 5 and start with “S”  *Smith, Smith, Sully* | Select Last\_Name from Employees  where Length(Last\_Name)=5 and Last\_Name like 'S%'; |
| List out the employees who are working in department 10 and draw the salaries more than 3500  *Jennifer Whalen* | Select First\_Name, Last\_Name from Employees  where Department\_ID = 10 and Salary >3500; |
| List out the employees who are not receiving commission (arranged alphabetically by Last Name)  *72 rows* | Select Last\_Name, First\_Name from Employees  where Commission\_PCT is NULL  Order by Last\_Name; |
| List out the employee id, last name in ascending order based on the employee id.  *107 rows* | Select Employee\_ID, Last\_Name from Employees  Order by Employee\_ID ASC; |
| List out the employee id, name in descending order based on salary column  *107 rows* | Select Employee\_ID, Last\_Name, First\_Name from Employees  Order by Salary; |
| List out the employee details according to their last\_name in ascending order and salaries in descending order  *107 rows* | Select \* from Employees  Order by Last\_Name Asc, Salary Desc; |
| List out the employee details according to their last\_name in ascending order and then on department\_id in descending order.  *107 rows* | Select \* from Employees  Order by Last\_Name Asc, Department\_ID Desc; |
| How many employees who are working in each different department in the organization  *28 rows* | Select d.Department\_Name "Department",  Count(e.Employee\_ID) "Count"  from Employees e Full Join Departments d  On d.Department\_ID = e.Department\_ID  Group by d.Department\_Name  Order by Count(e.Employee\_ID) Desc |
| List out the department wise maximum salary, minimum salary, average salary of the employees  *10 rows* | Select Department\_ID, Round(Min(Salary)) as "MIN", Round(Max(Salary)) as "MAX", Round(AVG(Salary)) as "AVG"  from Employees  Group by Department\_ID |
| List out the job wise maximum salary, minimum salary, average salaries of the employees.  *19 rows* | Select Job\_ID,  Round(MAX(Salary)) as "Max", Round(MIN(Salary)) as "Min", Round(AVG(Salary)) as "Average"  from Employees  Group by Job\_ID |
| List out the no.of employees joined in every month in ascending order.  *12 rows* | Select Count(Employee\_ID) as "No. of Employees", To\_Char(Hire\_Date,'Mon') as "Monthly Hires" from Employees  Group by to\_char(Hire\_Date,'Mon')  Order by Count(Employee\_ID) Asc; |
| List out the no.of employees for each month and year, in the ascending order based on the year, month.  *53 rows* | Select Count(Employee\_ID), To\_Char(Hire\_Date, 'Mon-YY') from Employees  Group by To\_Char(Hire\_Date, 'Mon-YY')  Order by To\_Char(Hire\_Date, 'Mon-YY') Asc; |
| List all the department ids having at least four employees.  *5 rows* | Select Department\_ID, Count(Department\_ID) as "No of Employees"  from Employees  Group by Department\_ID Having Count(Department\_ID)>=4 |
| How many employees joined in the month of January  *14* | Select Count(Employee\_ID)  as "Jan Total Hire"  from Employees  Where To\_Char(Hire\_Date, 'Mon-YY') like '%Jan%' |
| How many employees who are joined in January or September month.  *19* | Select Count(Employee\_ID) as "Jan & Sept Hire"  from Employees  Where To\_Char(Hire\_Date, 'Mon-YY') like '%Jan%'  or To\_Char(Hire\_Date, 'Mon-YY') like '%Sep%' |
| How many employees who are joined in 2006.  *24* | Select Count(Employee\_ID)  as "2006 Total Hire"  from Employees  Where To\_Char(Hire\_Date, 'Mon-YY') like '%06' |
| How many employees joined each month in 2006. | Select Hire\_Date, Count(Employee\_ID) from Employees  Where To\_Char(Hire\_Date, 'Mon-YY') Like '%-06'  GROUP BY Hire\_Date  Order by Hire\_Date |
| How many employees who are joined in March 2006. | Select Hire\_Date, Count(Employee\_ID)  from Employees  Where To\_Char(Hire\_Date,'Mon-YY') Like 'Mar-06'  GROUP BY Hire\_Date |
| Which department id is having greater than or equal to 2 employees joined in April 2006. | SELECT DEPARTMENT\_ID,COUNT(DEPARTMENT\_ID)  FROM EMPLOYEES  WHERE TO\_CHAR(HIRE\_DATE,'YYYY')=2006  GROUP BY DEPARTMENT\_ID  HAVING COUNT(DEPARTMENT\_ID)>=2  ORDER BY DEPARTMENT\_ID |
| Display the countries from the countries table, but display them only once.(use distinct) | Select Distinct(Country\_Name) from Countries; |
| How many employees joined in the year 2005  *29* | Select Count(Employee\_ID) "2005 Hired" from Employees  Where Hire\_Date like '%2005' |
| Display how many employees joined after 15th of the month.  ✅ | SELECT COUNT(EMPLOYEE\_ID) FROM EMPLOYEES  WHERE TO\_CHAR(HIRE\_DATE,'DD') >15 |
| Display the employees who are working in “Oxford” (should use sub query) | SELECT EMPLOYEES.EMPLOYEE\_ID,  EMPLOYEES.FIRST\_NAME,  EMPLOYEES.LAST\_N AME,  LOCATIONS.CITY FROM EMPLOYEES  JOIN  DEPARTMENTS ON DEPARTMENTS.DEPARTMENT\_ID=  EMPLOYEES.DEPARTMENT\_ID  JOIN  LOCATIONS ON LOCATIONS.LOCATION\_ID=  DEPARTMENTS.LOCATION\_ID  WHERE LOCATIONS.CITY='Oxford' |
| Display daily pay of employee of department 100 truncated to the nearest dollar --(hint for one day pay formula is trunc(salary/30) Employees salary that you see is a monthy salarr. To get annual salary multiply with 12 and then to get a daily salary divide that by 365 | SELECT EMPLOYEE\_ID,FIRST\_NAME,LAST\_NAME,TRUNC(SALARY/365) FROM EMPLOYEES WHERE DEPARTMENT\_ID=100 |
| Display date in this format --08:10:19 01/07/2013 Which is 'hh:mi:ss mm/dd/yyyy' | SELECT TO\_CHAR(SYSDATE,'HH:MI:SS MM/DD/YYYY') FROM DUAL Sub-Queries |
| Display the details of the employee drawing the second highest salary | Select \* from employees where salary=(select max(salary) from employees where salary <(select max(salary) from employees)) |
| List Employee id ,last name and their department name for all employees | SELECT EMPLOYEES.EMPLOYEE\_ID,EMPLOYEES.LAST\_NAME,DEPARTMENTS.DEPA RTMENT\_NAME FROM EMPLOYEES JOIN DEPARTMENTS ON EMPLOYEES.DEPARTMENT\_ID=DEPARTMENTS.DEPARTMENT\_ID |
| Display employee id , lastname and their JOB\_TITLE(designation) | SELECT EMPLOYEES.EMPLOYEE\_ID,EMPLOYEES.LAST\_NAME,JOBS.JOB\_TITLE FROM EMPLOYEES JOIN JOBS ON EMPLOYEES.JOB\_ID=JOBS.JOB\_ID |
| Display the employees with their department name and city. | SELECT EMPLOYEES.EMPLOYEE\_ID,EMPLOYEES.FIRST\_NAME,EMPLOYEES.LAST\_N AME,DEPARTMENTS.DEPARTMENT\_NAME,LOCATIONS.CITY FROM EMPLOYEES JOIN DEPARTMENTS ON EMPLOYEES.DEPARTMENT\_ID=DEPARTMENTS.DEPARTMENT\_ID JOIN LOCATIONS ON LOCATIONS.LOCATION\_ID=DEPARTMENTS.LOCATION\_ID |
| List the department names and get the count of employees working in each department | SELECT COUNT(EMPLOYEES.EMPLOYEE\_ID),DEPARTMENTS.DEPARTMENT\_NAME FROM EMPLOYEES JOIN DEPARTMENTS ON EMPLOYEES.DEPARTMENT\_ID=DEPARTMENTS.DEPARTMENT\_ID GROUP BY DEPARTMENTS.DEPARTMENT\_NAME |
| How many employees are working in sales department.? | SELECT COUNT(EMPLOYEES.EMPLOYEE\_ID),DEPARTMENTS.DEPARTMENT\_NAME FROM EMPLOYEES JOIN DEPARTMENTS ON EMPLOYEES.DEPARTMENT\_ID=DEPARTMENTS.DEPARTMENT\_ID GROUP BY DEPARTMENTS.DEPARTMENT\_NAME HAVING DEPARTMENTS.DEPARTMENT\_NAME ='Sales' |
| List the departments having greater than or equal to 5 employees and display the department names in ascending order. | SELECT COUNT(EMPLOYEES.EMPLOYEE\_ID),DEPARTMENTS.DEPARTMENT\_NAME FROM EMPLOYEES JOIN DEPARTMENTS ON EMPLOYEES.DEPARTMENT\_ID=DEPARTMENTS.DEPARTMENT\_ID GROUP BY DEPARTMENTS.DEPARTMENT\_NAME HAVING COUNT(EMPLOYEES.EMPLOYEE\_ID)>=5 ORDER BY DEPARTMENTS.DEPARTMENT\_NAME ASC |
| How many employees are there for each job\_title (designation) | SELECT COUNT(EMPLOYEES.EMPLOYEE\_ID),JOBS.JOB\_TITLE FROM EMPLOYEES,JOBS WHERE JOBS.JOB\_ID=EMPLOYEES.JOB\_ID GROUP BY JOBS.JOB\_TITLE |
| Display employee ID , employee last name and department id for employees who did more than one job in the past.(use job\_history table) --clue(join job history and employees table) | SELECT JOB\_HISTORY.EMPLOYEE\_ID,EMPLOYEES.LAST\_NAME,EMPLOYEES.DEPAR TMENT\_ID FROM EMPLOYEES JOIN JOB\_HISTORY ON EMPLOYEES.EMPLOYEE\_ID=JOB\_HISTORY.EMPLOYEE\_ID WHERE JOB\_HISTORY.EMPLOYEE\_ID IN (SELECT JOB\_HISTORY.EMPLOYEE\_ID FROM JOB\_HISTORY GROUP BY JOB\_HISTORY.EMPLOYEE\_ID HAVING COUNT(\*)>=2); |
| Display the employee details who earn more than their managers salaries. --51. show the count of employees under a manager ( this is example for self join) Use the employees table twice in the select clause | SELECT E1.MANAGER\_ID,COUNT(E1.EMPLOYEE\_ID) FROM EMPLOYEES E1,EMPLOYEES E2 WHERE E1.EMPLOYEE\_ID =E2.EMPLOYEE\_ID GROUP BY E1.MANAGER\_ID ORDER BY MANAGER\_ID |
| Display employee details for all departments (even if there is no employee in a department. | SELECT DEPARTMENTS.DEPARTMENT\_ID,EMPLOYEES.FIRST\_NAME,EMPLOYEES.L AST\_NAME FROM EMPLOYEES FULL OUTER JOIN DEPARTMENTS ON DEPARTMENTS.DEPARTMENT\_ID=EMPLOYEES.DEPARTMENT\_ID |
| Display all Employess in Sales & Purchasing departments | SELECT EMPLOYEES.EMPLOYEE\_ID,EMPLOYEES.FIRST\_NAME,EMPLOYEES.LAST\_N AME,DEPARTMENTS.DEPARTMENT\_NAME FROM EMPLOYEES LEFT JOIN DEPARTMENTS ON EMPLOYEES.DEPARTMENT\_ID=DEPARTMENTS.DEPARTMENT\_ID WHERE DEPARTMENTS.DEPARTMENT\_NAME in('Sales','Purchasing') |
| List distinct job\_title from jobs table for employees whose department names are Sales and AccountingDepartments. | SELECT DISTINCT JOB\_TITLE,DEPARTMENTS.DEPARTMENT\_ID,DEPARTMENTS.DEPARTMENT \_NAME FROM EMPLOYEES JOIN DEPARTMENTS ON EMPLOYEES.DEPARTMENT\_ID =DEPARTMENTS.DEPARTMENT\_ID JOIN JOBS ON JOBS.JOB\_ID =EMPLOYEES.JOB\_ID WHERE DEPARTMENTS.DEPARTMENT\_name in('Sales','Accounting'); |
| Syntax for instr is INSTR (string, character[ or substring], position, occurrence) --RETURNs a NUMBER --Output of below query is what? | SELECT INSTR('CORPORATE FLOOR','OR', 3, 2) FROM DUAL; SELECT INSTR('CORPORATE FLOOR','OR', -3, 2) FROM DUAL; |
| There STATE\_PROVINCE column values that are null in the locations table. Write a query to display values as N/A where there is null in the STATE\_PROVINCE field. ---NVL function lets you substitute a value when a null value is encountered. ----Example : SELECT NVL(alphabets, 'XXX')FROM onetable; | SELECT LOCATION\_ID, STATE\_PROVINCE, NVL(NULL,'XXX') FROM LOCATIONS |
| Where alphabets Is the column name and onetable is the table name | SELECT LOCATION\_ID, STATE\_PROVINCE, NVL(NULL,'XXX') FROM LOCATIONS |
| Display job ID, number of employees, sum of salary of each job id, and difference between highest salary and lowest salary of the employees belong to each job id. | SELECT JOB\_ID,COUNT(EMPLOYEE\_ID),SUM(SALARY),MAX(SALARY)- MIN(SALARY) AS "DIFFERENCE" FROM EMPLOYEES GROUP BY JOB\_ID |
| Display manager ID and number of employees managed by the manager. SELECT MANAGER\_ID,COUNT(EMPLOYEE\_ID)AS"NO OF EMPLOYEES" FROM EMPLOYEES GROUP BY MANAGER\_ID | List all the countries starting with ‘A’ from the countries table SELECT \* FROM COUNTRIES WHERE COUNTRY\_NAME LIKE 'A%' |
| In Oracle there is a facility to restrict the no of rows while showing output. It is done Using ROWNUM. ROWNUM is a Pseudocolumn | SELECT \* FROM EMPLOYEES WHERE ROWNUM<=50 |

| In [statistics,](https://docs.google.com/spreadsheets/d/1rKva_g2gBWglRqLnCJj2ue5tSXnKD3O7PV6cgwx74L0/edit?usp=sharing) percentiles are used to understand and interpret data. The *n*th percentile of a set of data is the value at which *n* percent of the data is below it. What you need to find out is the salary percentile for employee with the name David Austin, who is currently making 4800. You need to find out the salary range for his position and then calculate (in SQL) his percentile. If it turns out that he earns min salary for the position then the percentile is 0, if max salary - 100%.  *0.5* | Select Job\_ID, Salary FROM Employees  Where First\_Name = 'David' and Last\_Name = 'Austin'  IT\_PROG, 4800  Select MIN\_SALARY, MAX\_SALARY from Jobs  Where Job\_ID = 'IT\_PROG'  Min and Max Salary (4000-10,000)  Select SALARY FROM Employees  Where Job\_ID = 'IT\_PROG'  ORDER by SALARY ASC  4000, 4200, 4800, 4800, 6000, 9000, 10,000  Percentile = (Number of Values Below “x” / Total Number of Values) × 100  **42.86** = (3/7) X100  Or (FROM EMPLOYEES table ONLY)  SELECT first\_name, last\_name, salary, department\_id,  PERCENT\_RANK()  OVER (PARTITION BY department\_id ORDER BY salary DESC) "Percent\_Rank"  FROM employees  WHERE department\_id=60; |
| --- | --- |
| Calculate ratio of avg salaries in IT dept and Sales dept.  *0.64* | Select  Round((Select AVG(Salary) from Employees  Where Department\_ID IN (Select Department\_ID From Departments  Where Department\_Name = 'IT'))  /  (Select AVG(Salary) from Employees  where Department\_ID IN (Select Department\_ID From Departments  Where Department\_Name = 'Sales')),2) as "Ratio of Average Employees"  From Employees  where rownum=1; |
| Find out number of employees in all departments except IT and Administration.  *101* | Select Count(Employee\_ID) from Employees  Where Department\_ID IN (Select Department\_ID from Departments  Where Department\_Name <> 'IT' AND  Department\_Name <> 'Administration') OR Department\_ID is NULL; |
| Company plans to hire Public Relations Representative and pay medium salary for that position. Find out that number.  *10,000* | SELECT MEDIAN(salary) FROM employees  Where Job\_ID IN (Select Job\_ID from jobs  Where Job\_Title = 'Public Relations Representative'); |
| Where department NOC is located?  *2004 Charade Rd, Seattle, Washington, 98199*  *US* | Select Street\_Address, City, State\_Province, Postal\_Code, Country\_ID from Locations  Where Location\_ID IN (Select Location\_ID From Departments  Where Department\_Name = 'NOC'); |
| How many countries are in region 1?  *8* | Select Count(Country\_Name) from Countries  Where Region\_ID = 1; |
| Find out year with biggest number of hired employees  *2005* | Select \* From  (SELECT EXTRACT(year FROM Hire\_Date) "Year",  COUNT(Hire\_date) "No. of Hires"  FROM Employees  GROUP BY EXTRACT (year FROM Hire\_Date)  ORDER BY "No. of Hires" DESC)  where rownum=1; |