**ASSIGNMENT 5: SQL**

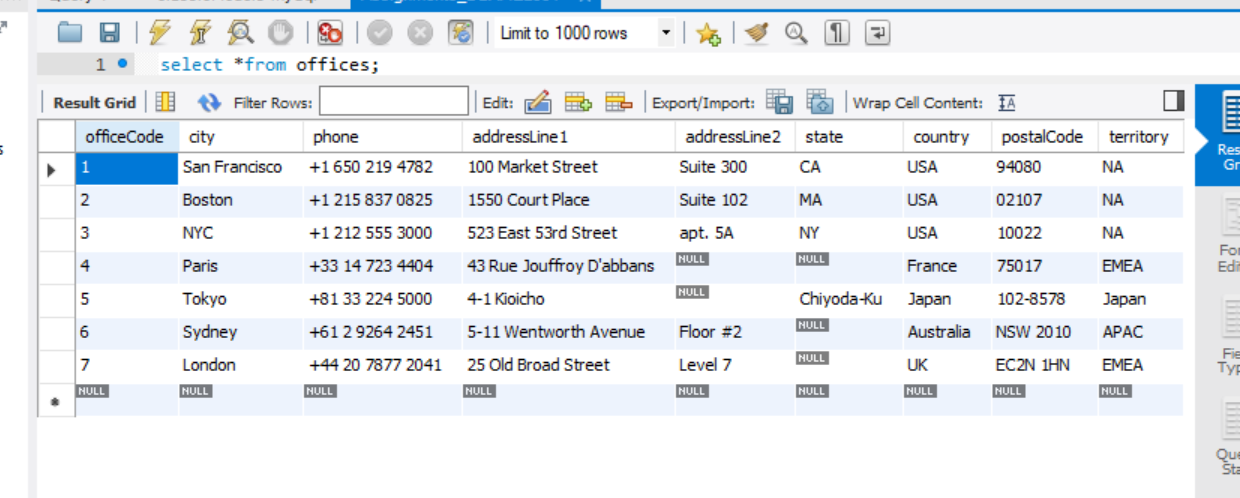
Francis Eseko

ESKFRA001

The Queries are as listed below, with the corresponding code and results obtained when running.

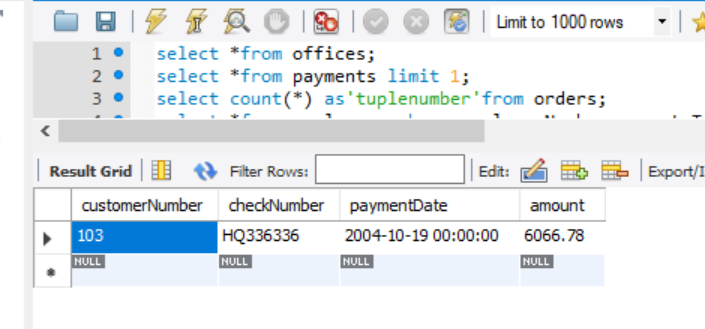
1. Show all information in the **offices** relation.

SQL code: select \* from offices;



1. Show any one tuple in the **payments** relation (just one, no more).

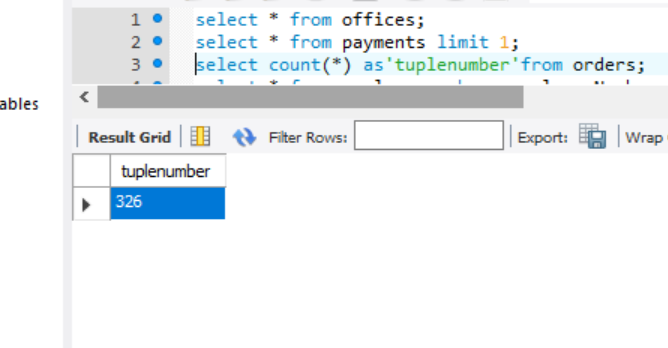
SQL code: select \* from payments limit 1;



1. Show how many tuples there are in the **orders** relation.

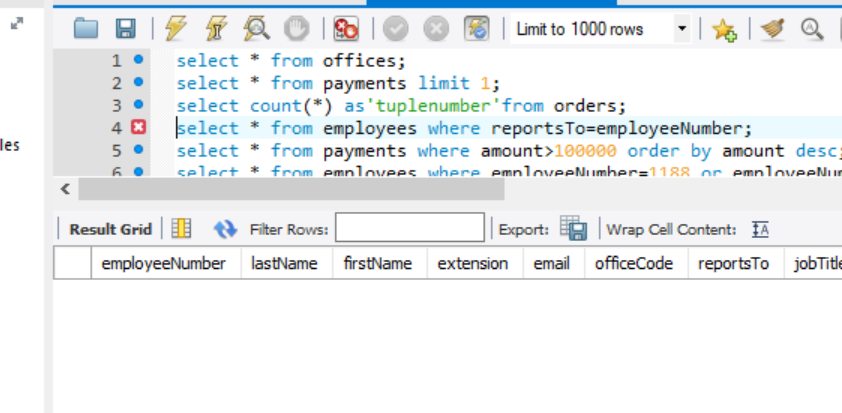
SQL code: select count(\*) as ‘tuplenumber’ from orders;

Results: 326 tuples



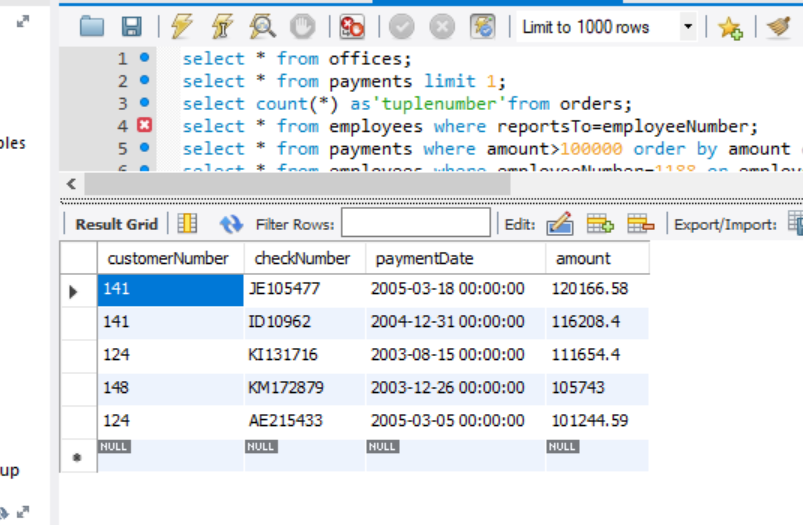
1. Show all **employees** tuples where **reportsTo** is the same as **employeeNumber**.

SQL code: select \* from employees where employeeNumber=reportsTo;



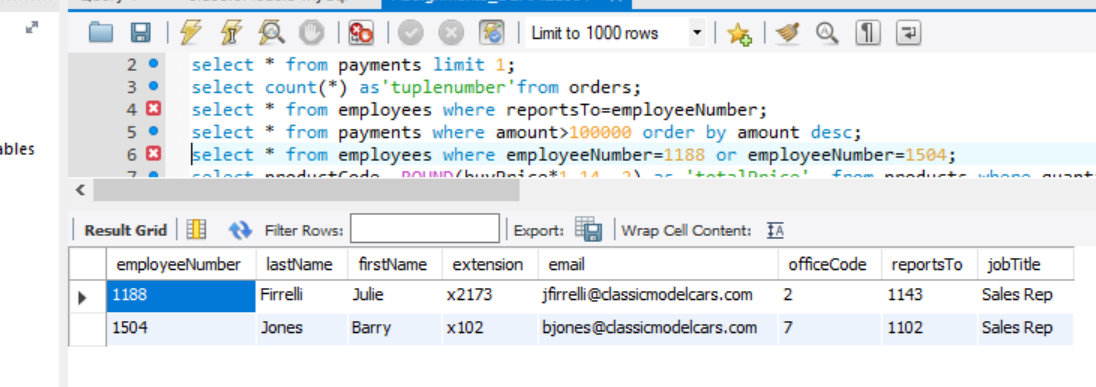
1. Show all information in the **payments** relation for payments exceeding 100 000, in decreasing order (i.e. from highest payment downwards).

SQL code: select \* from payments where amount>100000 order by amount desc;



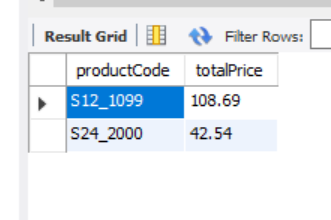
1. Show all information in the **employees** relation for **employeeNumber**s1188 and 1504.

SQL code: select \* from employees where employeeNumber=1188 or employeeNumber=1504;



1. Show the **productCode** of all **products** having their **quantityInStock** below 100, along with their total price. The total price is the **buyPrice**  plus VAT (VAT is 14% of **buyPrice**).

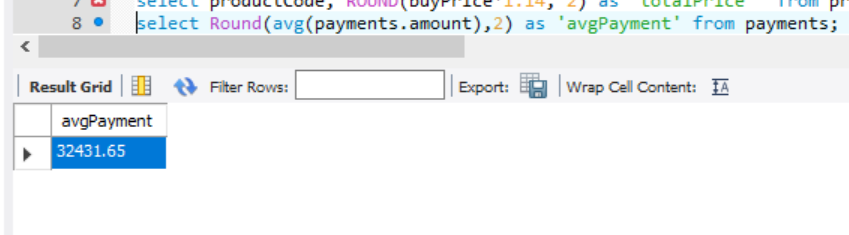
SQL code: select productCode, ROUND(buyPrice\*1.14, 2) as ‘totalPrice’ from products where quantityInStock<100;



1. What is the average **payment amount** in the database?

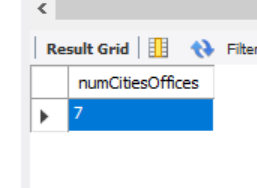
SQL code: select Round(avg(payments.amount),2) as ‘avgPayment’ from payments;

Rounded off inorder to simplify answer having multiple decimal points.



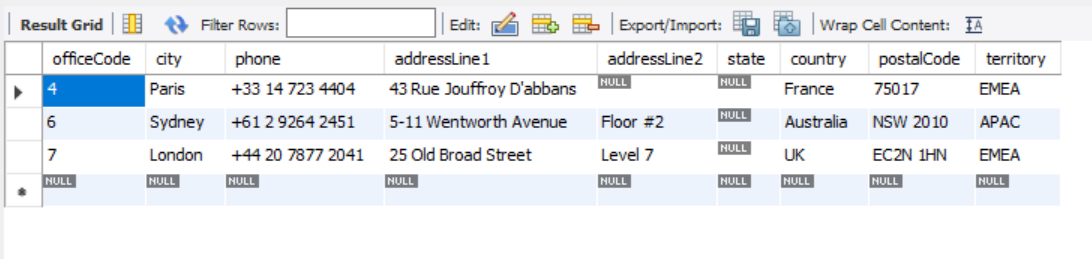
1. In how many cities are **offices** located (how many cities have **offices** in them)?

SQL code: select count(distinct city) as ‘numCitiesOffices’ from offices;



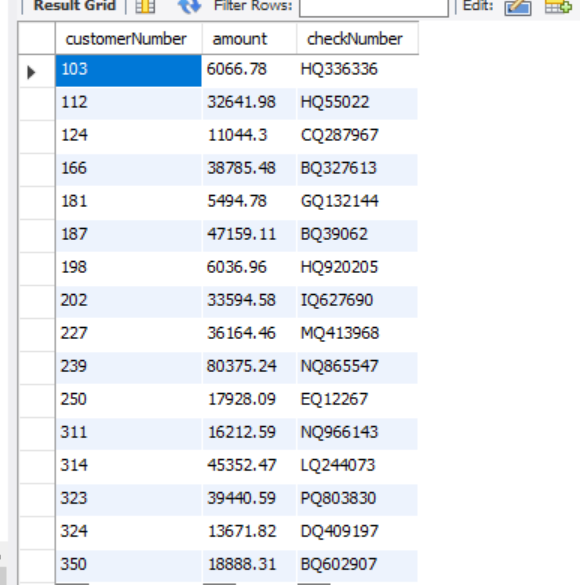
1. Show all information in the **offices** relation where the **state** is missing/unknown.

SQL code: select \* from offices where state is null;



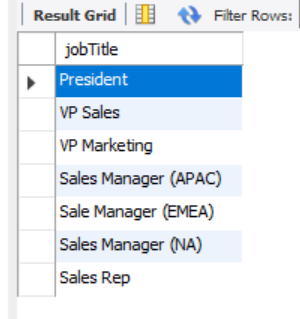
1. Show the **customerNumber** and **amount** for all **payments**  with a ‘Q’ as the 2nd character of the **checkNumber** (a check is a cheque!)

SQL code: select customerNumber, amount, checkNumber from payments where checkNumber like ‘\_Q%’;



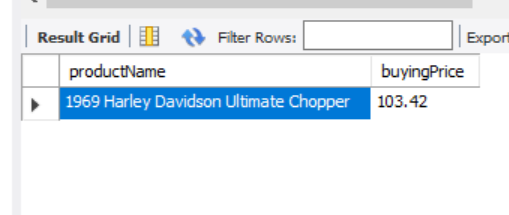
1. What **jobTitle**s exist in the database?

SQL code: select distinct(jobTitle) from employees;



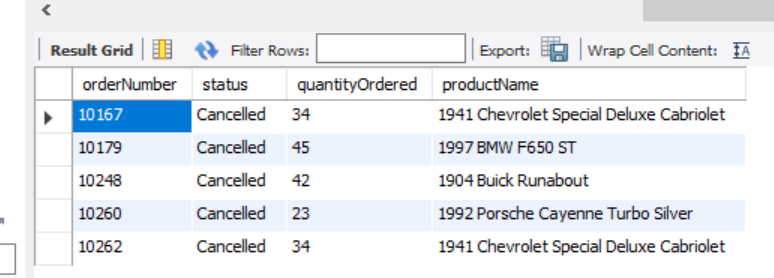
1. Show **productName** and **buyPrice** of the product(s) with the highest **buyPrice**.

SQL code: select productName, Max(buyPrice) as buyingPrice from products;



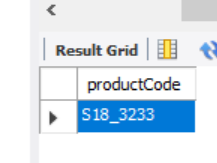
1. Show **orderNumber, status, quantityOrdered** and **productName** for all **products** from **productVendor** ‘Exoto Designs’ that have **status** ‘Cancelled’.

SQL code: select o.orderNumber, o.status, pr.quantityOrdered, p.productName from orders as o, products as p, orderdetails as pr where pr.orderNumber=o.orderNumber and p.productCode=pr.productCode and o.status=’Cancelled’ and p.productVendor=’Exoto Designs’;



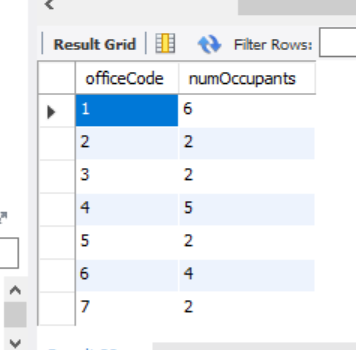
1. Show the **productCode** of all **products** that have never been ordered.

SQL code: select p.productCode from products as p left join orderdetails as pr on pr.productCode=p.productCode where pr.productCode is null;



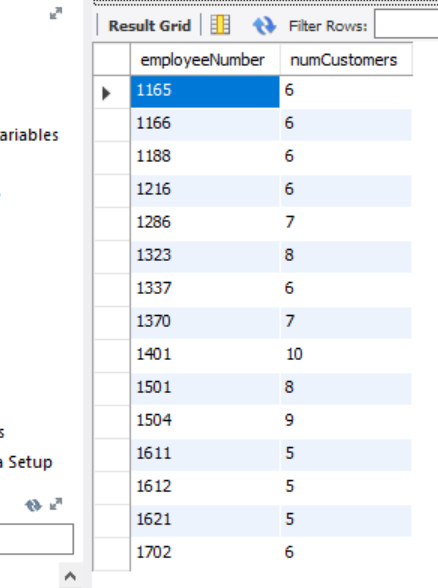
1. Show how many **employees** there are in each office (give **officeCode** and value each time).

SQL code: select o.officeCode, count(e.officeCode) as 'numOccupants' from offices as o, employees as e where e.officeCode=o.officeCode group by o.officeCode;



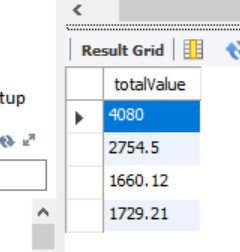
1. Show how many **customers** each employee is associated with (as **salesRepEmployeeNumber**), but only for employees whoare the **salesRepEmployeeNumber** for at least 1 customer. Give **employeeNumber** and value each time.

SQL code: select e.employeeNumber, count(c.salesRepEmployeeNumber) as 'numCustomers' from employees as e, customers as c where c.salesRepEmployeeNumber=e.employeeNumber group by e.employeeNumber;



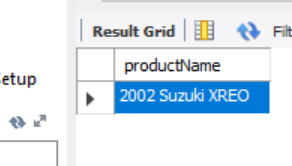
1. What was the total value of **orderNumber** 10100 i.e. the total of (**quantityOrdered \* priceEach)** over all its orderlines?

SQL code: select (pr.quantityOrdered\*pr.priceEach) as 'totalvalue' from orderdetails as pr where pr.orderNumber='10100';



1. Show the **productName** of the product/s with the largest **quantityInStock**.

SQL code: select productName from products where quantityInStock = (select Max(quantityInStock) from products) group by productName;



1. Show the **employeeNumber** of **employees** who **reportsTo** the same person as does **employeeNumber** 1313 (i.e. who have the same boss as 1313).

SQL code: select e.employeeNumber from employees as e, (select \* from employees as ee where ee.employeeNumber=1313) as bb where e.reportsTo=bb.reportsTo;

1. Show the **employeeNumber** of all **employees** who are superiors of **employeeNumber** 1313 (i.e. the person 1313 **reportsTo**, and the employee who that person **reportsTo**, ... all the way up)
2. Devise a useful query of your own involving the most interesting usage of SQL you can think of. Explain clearly in a comment what it is meant to find from the database. Also explain how you know the SQL for this query is correct (showing intermediate results if necessary). Marks here will be proportional to the complexity and usefulness of the query you implement.
3. Add a new **office** to the database, giving it **officeCode** 999 (meaning planned for later). This office will be in Cape Town, but no other details are known yet. Make **state** ‘Western Province’.

SQL code: insert into offices (officeCode, city , state) values(999, 'Cape Town', 'Western Cape');

1. Employee 1313 is superstitious. Change their employee number in the database, giving them the employee number 1 greater than the largest employee number in the database.
2. **OrderNumber** 10101 was never signed by the customer. Remove it from the database.