Sebastian Valenzuela Urena

SQL Queries

Purpose

The primary goal of this sample is to demonstrate different scenarios where SQL queries are created to filter, extract and create reports and databases for different useful purposes.

NOTE: *This is just an illustrative sample to demonstrate abilities of query creation. To observe the outcome of these queries it is necessary to have the data.*

Queries

```
-- Show all info of films

select *
from film;

-- Create a report that lists customer number, name, sales rep first and last name, order number, order comments, shipped date and required date for all orders where the order was shipped after the required date given by the customer.

select c.customerNumber, c.customerName, concat_ws(" ", e.firstName, e.lastName) as "Sales Representant", o.orderNUmber, o.comments, o.shippedDate, o.requiredDate
from customers c, employees e, orders o
where e.employeeNumber = c.salesRepEmployeeNumber
and c.customerNumber = o.customerNumber
and o.shippedDate > o.requiredDate;

-- Create a report  which lists all the products that are ordered on a Monday. (Different alternatives)

select distinct p.productName, dayname(o.orderDate) as "day name"
from products p, orders o, orderdetails od
where p.productCode = od.productCode
and o.orderNumber = od.orderNumber
and DAYNAME(orderDate) = 'MONDAY';

select productName
```

```
from products
where productCode in (select distinct od.productCode
                                                    from orders o, orderdetails od
                                                                where o.orderNumber =
                                                                        od.orderNumber
                                                            and DAYNAME(o.orderDate) =
                                                                        'MONDAY');


-- Can you create a report that lists all the products purchased by
-- 'Herkku Gifts'

select productName
from products
where productCode in (select distinct od.productCode
                                                    from orders o, orderdetails od,
Customers c
                                                    where o.orderNumber =
od.orderNumber
                    and o.CustomerNumber = c.customerNumber
                    and c.CustomerName = "Herkku Gifts");

-- Can you create a report that shows total payments for customer
-- "Atelier graphique"

select c.customerName, sum(p.amount)
from customers c, payments p
where c.customerNumber = p.customerNumber
and c.customerName = "Atelier graphique";

-- Show first name and last name of actors
select first_name, last_name
from actor;

-- Show first name, last name and email for customers
select first_name, last_name, email
from customer;

-- Show the unique values of rating in films
select distinct rating
from film;
```

```
--Count all the unique values for the amount of payments
select count (distinct amount)
from payment;

-- Show all the info for customers which first name is Jared
select *
from customer
where first_name = 'Jared';

-- Show the films which rental rate is greater than 4 and their replacement cost is
greater or equal to 19.99
select *
from film
where rental_rate > 4
and replacement_cost >= 19.99;

select count (*)
from film
where rating = 'R'
or rating = 'PG-13';

--Show all the info of all films which rating is not R
select count (*)
from film
where rating != 'R';

-- Find the email of the customer named Nancy Thomas
select email
from customer
where first_name = 'Nancy'
and last_name = 'Thomas';

-- A customer wants to know the description of the film Outlaw Hanky
select description
from film
where title = 'Outlaw Hanky';

--Show all info of customers sorted by first name ascending
select *
from customer
```

```
order by first_name asc;

select *
from customer
order by first_name asc, last_name desc;

--What is the customer id for the first 10 customers who made a payment?
select customer_id
from payment
order by payment_date
limit 10;

--Customer wants to know the titles for the 5 shortest movies
select title
from film
order by length asc
limit 5;

-- The previous customer wants to know how many films are less than 50 min
select count (title)
from film
where length <= 50;

-- What are the payments between 2007-02-01 and 2007-02-15
select *
from payment
where payment_date between '2007-02-01' and '2007-02-15';

-- How many payments have an amount that are not .99, 1.98, 1.99
select count (*)
from payment
where amount
not in (0.99, 1.98, 1.99);

--Select customers that has 'er' in their names
select *
from customer
where first_name
like '%er';

--Select the minimum and maximum replacement cost of the films
```

```
select min(replacement_cost), max(replacement_cost)
from film;

--Select the average replacement cost of the films rounded in 2 decimals
select round(avg(replacement_cost),2)
from film;

--Show the total amount of payments by customer id and staff id order by this total
select staff_id, customer_id, sum(amount)
from payment
group by staff_id, customer_id
order by sum(amount);

-- We want to give a bonus to the staff 1 or 2 that have the most amount of payments as number no dollar
select staff_id, count (payment_id)
from payment
group by staff_id;

--What is the average replacement cost per MPAA rating
select rating, round(avg(replacement_cost),2)
from film
group by rating;

--What are the customers ids for customers that spend more than 100 in payments transactions with staff_id 2
select customer_id, sum(amount)
from payment
where staff_id = 2
group by customer_id
having sum(amount) > 100;

--What customer has the highest customer ID number whose name starts with an 'E' and has an address ID lower than 500?
SELECT first_name,last_name FROM customer
WHERE first_name LIKE 'E%'
AND address_id <500
ORDER BY customer_id DESC
LIMIT 1;

--Inner Join
```

```
select payment_id, payment.customer_id, first_name
from payment
inner join customer
on payment.customer_id = customer.customer_id;

-- Full Outer Join
-- All payments are associated with a customer id, no customer without historical payment
--The results show us that there is not payment information not associated with a customer or a customer without a payment history
select *
from customer
full outer join payment
on customer.customer_id = payment.customer_id
where customer.customer_id is null
or payment.payment_id is null;

--Left Outer Join
-- Which movies are not in an inventory and store
select film.film_id, title, inventory_id, store_id
from film
left join inventory
on inventory.film_id = film.film_id
where inventory.film_id is null;

--What movies Nick Walhberg has been
select  film.title, actor.first_name, actor.last_name
from actor, film, film_actor
where actor.actor_id = film_actor.actor_id
and film_actor.film_id = film.film_id
and actor.first_name = 'Nick'
and actor.last_name = 'Wahlberg';

--Same question using Inner Join
select title, first_name, last_name
from actor
inner join film_actor
on actor.actor_id = film_actor.actor_id
inner join film
on film_actor.film_id = film.film_id
where first_name = 'Nick'
```

```
and last_name = 'Wahlberg';

--Extract the year of the payments date
select extract(year from payment_date)
from payment;

-- Change the date type to months and years in payment date
select to_char(payment_date, 'MONTH-YYYY')
from payment;

-- During which months did payments occur
select distinct to_char(payment_date, 'MONTH')
from payment;

--How many paymentsocurred on monday
select count(*)
from payment
where extract(dow from payment_date) = 1;

-- Create emails using the fisrt letter of the name, the last name and gmail format
select lower(left(first_name,1)) || lower(last_name) || '@gmail.com'
from customer;

--Subquery. what are the movies that have a greater rental rate than the average
rental rate
select title, rental_rate
from film
where rental_rate > (select avg(rental_rate)
                                        from film);

--Subquery
select film_id, title
from film
where film_id in
(select inventory.film_id
from rental
inner join inventory on inventory.inventory_id = rental.inventory_id
where return_date between '2005-05-29' and '2005-05-30')
order by title;
```

```
--Select the forst name and last name of cusotmer that payment amount is greater than 11
select first_name, last_name
from customer c
where exists
(select *
from payment p
where p.customer_id = c.customer_id
and amount > 11);

--Match the films that have the same lenght giving the titles of the movies that match in the same table
select f1.title, f2.title, f1.length
from film f1
inner join film f2
on f1.film_id != f2.film_id
and f1.length = f2.length
and f1.length = 117;

--Case
select customer_id,
case
        when customer_id <= 100 then 'Vip'
        when customer_id between 100 and 200 then 'Plus'
        else 'Normal'
end as "Category class"
from customer;

--Case expression
select customer_id,
case customer_id
        when 2 then 'Winner'
        when 5 then 'Looser'
        else 'Normal'
end as "Winner or Looser"
from customer;

--Count the number of movies that cost 0.99 using case
select
sum(case rental_rate
  when 0.99 then 1
```

```
  else 0
end) as bargains,
sum(case rental_rate
  when 2.99 then 1
  else 0
end) as normal
from film;
```