

Storage Technologies - P1

Feb 28, 2024

Uso de SQL

a) ¿Cuántos empleados hay?

```
1 SELECT count(emp_no) AS empleados_totales FROM EMPLOYEES
```

	empleados_totales bigint
1	71782

b) ¿Cuantos empleados hombres / mujeres hay en compañía?

```
1 SELECT gender, count(*) AS genero FROM EMPLOYEES GROUP BY gender
```

	gender gender	genero bigint
1	M	42985
2	F	28797

c) ¿Cuál es el nombre y apellido de los empleados contratados durante el año 2000?

```
1 SELECT first_name, last_name
2 FROM Employees
3 WHERE EXTRACT(YEAR FROM hire_date) = 2000
```

	first_name character varying (14)	last_name character varying (16)
1	Ulf	Flexer
2	Seshu	Rathonyi
3	Randi	Luit

d) ¿Cuántas veces se le ha subido el salario al empleado Bezalel Simmel desde que trabaja en la empresa?

```

1 WITH SalariosOrdenados AS (
2   SELECT emp_no, salary,
3         LEAD(salary) OVER (PARTITION BY emp_no ORDER BY from_date) AS salario_siguiente
4   FROM salaries
5   WHERE emp_no = (SELECT emp_no FROM Employees WHERE first_name = 'Bezalel' AND last_name = 'Simmel')
6 )
7 SELECT salary, salario_siguiente,
8        COUNT(CASE WHEN salario_siguiente > salary THEN 1 END) OVER () AS VecesAumento
9 FROM SalariosOrdenados;

```

	salary integer	salario_siguiente integer	vecesaumento bigint
1	65828	65909	5
2	65909	67534	5
3	67534	69366	5
4	69366	71963	5
5	71963	72527	5
6	72527	[null]	5

e) ¿Cuál es el salario máximo que ha percibido el empleado 10001?

```

1 SELECT MAX(salary) AS SalarioMaximo
2 FROM salaries
3 WHERE emp_no = 10001;

```

	salariomaximo integer
1	88958

f) ¿Cuál es la media de salario percibida por los trabajadores en el periodo comprendido entre 1990 y 1999? (no tengas en cuenta si a una misma persona se le ha subido el salario diversas veces en el periodo, puedes asumir que son personas distintas)

```

1 SELECT ROUND(AVG(salary), 2) AS MediaSalario1990s
2 FROM salaries
3 WHERE from_date >= '1990-01-01' AND to_date < '2000-01-01';

```

	mediasalario1990s numeric 🔒
1	62814.01

g) Lista todos los puestos existentes en la empresa ordenados alfabéticamente

```
1 SELECT DISTINCT title
2 FROM titles
3 ORDER BY title;
```

	title character varying (50) 🔒
1	Assistant Engineer
2	Engineer
3	Senior Engineer
4	Senior Staff
5	Staff
6	Technique Leader

h) ¿Cuál es el nombre y apellido de la persona que más cobra en la empresa? (una sola consulta)

```
1 SELECT e.first_name, e.last_name, s.salary
2 FROM employees e
3 JOIN salaries s ON e.emp_no = s.emp_no
4 ORDER BY s.salary DESC
5 LIMIT 1;
```

	first_name character varying (14) 🔒	last_name character varying (16) 🔒	salary integer 🔒
1	Itzhak	Ramaiah	145732

i) ¿Cuántos empleados había por departamento en el año 1990? (uso de la cláusula *inner join*). Ordena los resultados de mayor a menor.

```

1 SELECT d.dept_name, COUNT(e.emp_no) AS NumEmpleados
2 FROM departments d
3 INNER JOIN dept_emp de ON d.dept_no = de.dept_no
4 INNER JOIN employees e ON de.emp_no = e.emp_no
5 WHERE de.from_date <= '1990-12-31' AND (de.to_date > '1990-12-31' OR de.to_date IS NULL)
6 GROUP BY d.dept_name
7 ORDER BY NumEmpleados DESC;

```

	dept_name character varying (40) 🔒	numempleados bigint 🔒
1	Development	2370
2	Production	1934
3	Sales	1404
4	Customer Service	494
5	Quality Management	490
6	Research	487
7	Finance	465
8	Human Resources	458
9	Marketing	451

Modelado de datos

a) Implementar base de datos relacional.

Con el objetivo de tener control del catálogo de películas. Las entidades mínimas que tiene que contener el diseño se listan a continuación junto con sus atributos:

- Film (id, title, productionCo, director, genre)
- ProductionCo (id, name, country)
- Director (id, name, surname, country)
- Generes (id, name)

Diseño inicial para registrar información sobre películas, compañías de producción, directores y géneros de manera estructurada. Definiendo las claves primarias (PK) y las claves foráneas (FK).

❖ Film:

- id (PK): Identificador único de la película.
- title: Título de la película.
- productionCo_id (FK): Referencia al identificador único de la compañía de producción.
- director_id (FK): Referencia al identificador único del director.
- genre_id (FK): Referencia al identificador único del género.

- ❖ ProductionCo:
 - id (PK): Identificador único de la compañía de producción.
 - name: Nombre de la compañía de producción.
 - country: País de origen de la compañía de producción.
- ❖ Director:
 - id (PK): Identificador único del director.
 - name: Nombre del director.
 - surname: Apellido del director.
 - country: País de origen del director.
- ❖ Genres:
 - id (PK): Identificador único del género.
 - name: Nombre del género.

```
1  -- Crear la tabla ProductionCo
2  CREATE TABLE ProductionCo (
3      id INT PRIMARY KEY,
4      name VARCHAR(255) NOT NULL,
5      country VARCHAR(255) NOT NULL
6  );
7
8  -- Crear la tabla Director
9  CREATE TABLE Director (
10     id INT PRIMARY KEY,
11     name VARCHAR(255) NOT NULL,
12     surname VARCHAR(255) NOT NULL,
13     country VARCHAR(255) NOT NULL
14 );
15
16 -- Crear la tabla Genres
17 CREATE TABLE Genres (
18     id INT PRIMARY KEY,
19     name VARCHAR(255) NOT NULL
20 );
21
22 -- Crear la tabla Film
23 CREATE TABLE Film (
24     id INT PRIMARY KEY,
25     title VARCHAR(255) NOT NULL,
26     productionCo_id INT,
27     director_id INT,
28     genre_id INT,
29     FOREIGN KEY (productionCo_id) REFERENCES ProductionCo(id),
30     FOREIGN KEY (director_id) REFERENCES Director(id),
31     FOREIGN KEY (genre_id) REFERENCES Genres(id)
```

b) Inserción de resultados

Una vez creadas las tablas, insertar resultados inventados en cada una de ellas (mínimo 5 en cada una). Los nombres de las películas, directores, productora... los puedes sacar de internet.

```
1  -- Insertar datos en ProductionCo
2  INSERT INTO ProductionCo (id, name, country) VALUES
3  (1, 'ABC Productions', 'USA'),
4  (2, 'XYZ Studios', 'UK'),
5  (3, 'FilmCo International', 'Canada'),
6  (4, 'CineWorld', 'Spain'),
7  (5, 'Global Films', 'France');
8
9  -- Insertar datos en Director
10 INSERT INTO Director (id, name, surname, country) VALUES
11 (1, 'John', 'Doe', 'USA'),
12 (2, 'Jane', 'Smith', 'UK'),
13 (3, 'Michael', 'Johnson', 'Canada'),
14 (4, 'Sara', 'Lopez', 'Spain'),
15 (5, 'Pierre', 'Martin', 'France');
16
17 -- Insertar datos en Genres
18 INSERT INTO Genres (id, name) VALUES
19 (1, 'Action'),
20 (2, 'Drama'),
21 (3, 'Comedy'),
22 (4, 'Science Fiction'),
23 (5, 'Thriller');
24
25 -- Insertar datos en Film
26 INSERT INTO Film (id, title, productionCo_id, director_id, genre_id) VALUES
27 (1, 'The Adventure Begins', 1, 1, 1),
28 (2, 'Heartstrings', 2, 2, 2),
29 (3, 'Laugh Out Loud', 3, 3, 3),
30 (4, 'Galactic Odyssey', 4, 4, 4),
31 (5, 'Edge of Darkness', 5, 5, 5);
```

c) Actualización de los títulos

Haz una actualización en uno de los títulos de la película, por ejemplo, pon el título en mayúsculas.

```
1  UPDATE Film
2  SET title = UPPER(title)
3  WHERE id = 1;
```

ANNEXO CÓDIGO

a) ¿Cuántos empleados hay?

```
SELECT count(emp_no) AS empleados_totales FROM EMPLOYEES
```

b) ¿Cuántos empleados hombres / mujeres hay en compañía?

```
SELECT gender, count(*) AS genero FROM EMPLOYEES GROUP BY gender
```

c) ¿Cuál es el nombre y apellido de los empleados contratados durante el año 2000?

```
SELECT first_name, last_name  
FROM Employees  
WHERE EXTRACT(YEAR FROM hire_date) = 2000
```

d) ¿Cuántas veces se le ha subido el salario al empleado Bezalel Simmel desde que trabaja en la empresa?

```
WITH SalariosOrdenados AS (  
  SELECT emp_no, salary,  
         LEAD(salary) OVER (PARTITION BY emp_no ORDER BY from_date) AS  
  salario_siguiente  
  FROM salaries  
  WHERE emp_no = (SELECT emp_no FROM Employees WHERE first_name = 'Bezalel'  
AND last_name = 'Simmel')  
)  
SELECT salary, salario_siguiente,  
       COUNT(CASE WHEN salario_siguiente > salary THEN 1 END) OVER () AS  
VecesAumento  
FROM SalariosOrdenados;
```

e) ¿Cuál es el salario máximo que ha percibido el empleado 10001?

```
SELECT MAX(salary) AS SalarioMaximo  
FROM salaries  
WHERE emp_no = 10001;
```

f) ¿Cuál es la media de salario percibida por los trabajadores en el periodo comprendido entre 1990 y 1999? (no tengas en cuenta si a una misma persona se le ha subido el salario diversas veces en el periodo, puedes asumir que son personas distintas)

```
SELECT ROUND(AVG(salary), 2) AS MediaSalario1990s  
FROM salaries  
WHERE from_date >= '1990-01-01' AND to_date < '2000-01-01';
```

g) Lista todos los puestos existentes en la empresa ordenados alfabéticamente

```
SELECT DISTINCT title
FROM titles
ORDER BY title;
```

h) ¿Cuál es el nombre y apellido de la persona que más cobra en la empresa? (una sola consulta)

```
SELECT e.first_name, e.last_name, s.salary
FROM employees e
JOIN salaries s ON e.emp_no = s.emp_no
ORDER BY s.salary DESC
LIMIT 1;
```

i) ¿Cuántos empleados había por departamento en el año 1990? (uso de la cláusula inner join). Ordena los resultados de mayor a menor.

```
SELECT d.dept_name, COUNT(e.emp_no) AS NumEmpleados
FROM departments d
INNER JOIN dept_emp de ON d.dept_no = de.dept_no
INNER JOIN employees e ON de.emp_no = e.emp_no
WHERE de.from_date <= '1990-12-31' AND (de.to_date > '1990-12-31' OR de.to_date IS NULL)
GROUP BY d.dept_name
ORDER BY NumEmpleados DESC;
```

a) Implementar base de datos relacional.

```
-- Crear la tabla ProductionCo
CREATE TABLE ProductionCo (
  id INT PRIMARY KEY,
  name VARCHAR(255) NOT NULL,
  country VARCHAR(255) NOT NULL
);

-- Crear la tabla Director
CREATE TABLE Director (
  id INT PRIMARY KEY,
  name VARCHAR(255) NOT NULL,
  surname VARCHAR(255) NOT NULL,
  country VARCHAR(255) NOT NULL
);

-- Crear la tabla Genres
CREATE TABLE Genres (
  id INT PRIMARY KEY,
  name VARCHAR(255) NOT NULL
);

-- Crear la tabla Film
```



```
CREATE TABLE Film (  
  id INT PRIMARY KEY,  
  title VARCHAR(255) NOT NULL,  
  productionCo_id INT,  
  director_id INT,  
  genre_id INT,  
  FOREIGN KEY (productionCo_id) REFERENCES ProductionCo(id),  
  FOREIGN KEY (director_id) REFERENCES Director(id),  
  FOREIGN KEY (genre_id) REFERENCES Genres(id)
```

b) Inserción de resultados

```
-- Insertar datos en ProductionCo  
INSERT INTO ProductionCo (id, name, country) VALUES  
(1, 'ABC Productions', 'USA'),  
(2, 'XYZ Studios', 'UK'),  
(3, 'FilmCo International', 'Canada'),  
(4, 'CineWorld', 'Spain'),  
(5, 'Global Films', 'France');  
  
-- Insertar datos en Director  
INSERT INTO Director (id, name, surname, country) VALUES  
(1, 'John', 'Doe', 'USA'),  
(2, 'Jane', 'Smith', 'UK'),  
(3, 'Michael', 'Johnson', 'Canada'),  
(4, 'Sara', 'Lopez', 'Spain'),  
(5, 'Pierre', 'Martin', 'France');  
  
-- Insertar datos en Genres  
INSERT INTO Genres (id, name) VALUES  
(1, 'Action'),  
(2, 'Drama'),  
(3, 'Comedy'),  
(4, 'Science Fiction'),  
(5, 'Thriller');  
  
-- Insertar datos en Film  
INSERT INTO Film (id, title, productionCo_id, director_id, genre_id) VALUES  
(1, 'The Adventure Begins', 1, 1, 1),  
(2, 'Heartstrings', 2, 2, 2),  
(3, 'Laugh Out Loud', 3, 3, 3),  
(4, 'Galactic Odyssey', 4, 4, 4),  
(5, 'Edge of Darkness', 5, 5, 5);
```

c) Actualización de los títulos

```
UPDATE Film  
SET title = UPPER(title)  
WHERE id = 1;
```