Code examples from lectures SELECT Queries

SELECT Queries

1. **Retrieve top 10 most expensive products:**

```
SELECT product_name, unit_price
FROM products
ORDER BY unit_price DESC
LIMIT 10;
```

   - `ORDER BY`: Used to sort the result set in ascending or descending order.
   - `DESC`: Sorts the result in descending order.
   - `LIMIT`: Limits the number of rows returned.
   - Fetches `product_name` and `unit_price` of the top 10 most expensive products.

2. **Sum of freight charges by employee:**

```
SELECT employee_id, SUM(freight)
FROM orders
GROUP BY employee_id;
```

   - `SUM`: An aggregate function that returns the sum of a numeric column.
   - Fetches `employee_id` and the sum of `freight` charges grouped by each employee.

3. **City-wise average, maximum, and minimum age of employees in London:**

```
SELECT city,
  AVG(EXTRACT(year from AGE(CURRENT_TIMESTAMP, birth_date))),
  MAX(EXTRACT(year from AGE(CURRENT_TIMESTAMP, birth_date))),
  MIN(EXTRACT(year from AGE(CURRENT_TIMESTAMP, birth_date)))
FROM employees
WHERE city = 'London'
GROUP BY city;
```

   - `AVG`, `MAX`, `MIN`: Aggregate functions to calculate average, maximum, and minimum values respectively.
   - `CURRENT_TIMESTAMP`: Returns the current date and time.

   - Fetches `city`, average age, maximum age, and minimum age of employees in London.

4. **City-wise average age of employees above 60:**

```sql
SELECT city, AVG(EXTRACT(year from AGE(CURRENT_TIMESTAMP, birth_date))) AS avg_age
FROM employees
GROUP BY city
HAVING EXTRACT(year from AGE(CURRENT_TIMESTAMP, birth_date)) > 60;
```

   - **`HAVING`**: Used to filter records that work on aggregated data.
   - Fetches `city` and average age of employees whose age is above 60, grouped by `city`.

5. **Retrieve the oldest employee:**

```sql
SELECT first_name, last_name, EXTRACT(year from AGE(CURRENT_TIMESTAMP, birth_date)) AS age
FROM employees
ORDER BY age DESC
LIMIT 1;
```

   - Fetches `first_name`, `last_name`, and age of the oldest employee.

6. **Retrieve top 3 oldest employees:**

```sql
SELECT first_name, last_name, EXTRACT(year from AGE(CURRENT_TIMESTAMP, birth_date)) AS age
FROM employees
ORDER BY age DESC
LIMIT 3;
```

   - Fetches `first_name`, `last_name`, and age of the top 3 oldest employees.

7. **Formatted greeting with birth date:**

```sql
SELECT 'Dear ' || last_name ||' '||first_name AS welcome,
  'Your Birth Day is on ' || CAST(extract(month from birth_date) AS varchar(10)) ||'  ' ||CAST(extract(day from birth_date) AS varchar(10)) || 'th'
FROM employees;
```

- `||`: Concatenation operator in SQL.
- **`CAST`**: Converts a value from one data type to another.
- Creates a greeting message with the employee's name and birth date.

8. **Formatted greeting with birth date (alternative syntax):**

```
SELECT 'Dear ' || last_name ||' '||first_name AS welcome,
  'Your Birth Day is on ' || extract(month from birth_date)::varchar(10)
||'  ' ||extract(day from birth_date)::varchar(10) || 'th'
FROM employees;
```

- **`::`**: Type cast operator in PostgreSQL.
- Creates a greeting message with the employee's name and birth date.

9. **Concatenate product name and quantity per unit:**

```
SELECT product_name || quantity_per_unit AS name_and_quant, unit_price,
units_in_stock
FROM products;
```

- Concatenates `product_name` and `quantity_per_unit` into a single field and fetches `unit_price` and `units_in_stock`.

10. **Retrieve shortened product names:**

```
SELECT SUBSTRING(product_name, 1, 10) AS short_name, unit_price,
units_in_stock
FROM products;
```

- **`SUBSTRING`**: Extracts a substring from a string.
- Fetches the first 10 characters of `product_name`, along with `unit_price` and `units_in_stock`.

11. **Replace 'Tea' with 'Coffee' in product names:**

```
SELECT REPLACE(product_name, 'Tea', 'Coffee') AS changed_name, unit_price
FROM products
WHERE product_name LIKE '%Tea%';
```

- **`REPLACE`**: Replaces all occurrences of a specified string value with another string value.
- Fetches `product_name` with 'Tea' replaced by 'Coffee' for products whose names contain 'Tea', along with `unit_price`.

12. **Concatenate first and last name, and extract birth year:**

```sql
SELECT first_name || ' ' || last_name AS full_name,
  DATE_PART('year', birth_date) AS birth_year
FROM employees;
```

- **`DATE_PART`**: Extracts a subfield from a date/time value.
- Fetches concatenated `first_name` and `last_name`, along with the year part of `birth_date`.

13. **Concatenate first and last name, and calculate age:**

```sql
SELECT first_name || ' ' || last_name AS full_name,
  AGE(NOW(), birth_date) AS age
FROM employees;
```

- **`AGE`**: Calculates the age between two dates.
- **`NOW()`**: Returns the current date and time.
- Fetches concatenated `first_name` and `last_name`, along with the age calculated from `birth_date`.

14. **Concatenate first and last name, and extract birth month:**

```sql
SELECT first_name || ' ' || last_name AS full_name,
  EXTRACT(month from AGE(NOW(), birth_date)) AS birth_month
FROM employees;
```

- Fetches concatenated `first_name` and `last_name`, along with the month part of the age calculated from `birth_date`.