**CHAPTER II**

A close up of a text

Description automatically generated

* abgeschlossenes Studium eines MINT-Fachs (Hast Du dies nicht und bist trotzdem mit allen Anforderungen ausgestattet und motiviert, dann überzeuge uns gerne)
* sehr gutes Verständnis relationaler Datenbanken und Erfahrungen im Bau von ETL-Pipelines; Entwicklung mit Python oder Java; Organisation mit Docker, Airflow und dbt
* Erfahrungen in der Administration und Wartung von Linux-Servern und entsprechender Netzwerkkonfiguration
* praktische Erfahrung im Data Engineering und dem Bau von automatisierten Pipelines, vorzugsweise mit einem Mix aus: Python, Jenkins, (Postgre-) SQL oder Bash (zudem sind Erfahrungen mit gängigen Social Media APIs und Google- Schnittstellen, wie z.B. YouTube von Vorteil)
* Begeisterung für herausfordernde Problemstellungen und komplexe Datensätze sowie hierbei strukturiert, zielorientiert und eigenständig arbeiten zu können
* ein agiles Mindset und Freude an der Arbeit in einem diversen und internationalen Unternehmen
* technische Sachverhalte und komplexe Ergebnisse kannst Du Deinen DW Kolleg:innen verständlich vermitteln - und das in sehr gutem Deutsch und Englisch; für die Kollaboration bist Du sicher im Umgang mit Git

Data is Fact or Fi

-----------------------------------------------------------------------------------------------------------------------------------------------------------

-----------------------------------------------------------------------------------------------------------------------------------------------------------**CHAPTER II**

Filtering is to narrowing the data that we want to retrieve, and use when we want to analysis to reduce the number of records and also to increase query performance and also It reduces the strain on the client-side of the application

Clauses and Operators :

1. WHERE
2. BETWEEN
3. IN
4. OR
5. NOT
6. LIKE
7. ORDER BY
8. GROUP BY

Wildcards allow more precise search capabilities.

Aggregate Math Functions :

1. AVERAGE
2. COUNT
3. MAX
4. MIN

SELECT column\_name1, column\_name2 FROM table\_name WHERE column\_name operator value;

A screenshot of a computer

Description automatically generated

SELECT column\_name1, column\_name2 FROM table\_name WHERE condition 1 OR condition 2;

#DBMS will not evaluate the second condition in a WHERE clause if the first condition is met.

SELECT column\_name1, column\_name2 FROM table\_name WHERE column\_name IN (x,y,z) ;

BENEFITS of IN :

1. Long list of options, how many you can lists (multiple things).
2. IN Executes faster than OR
3. Don’t have to think about the order with IN
4. Can contain another SELECT clause/statement (nested)

A screenshot of a computer

Description automatically generated

SQL Process AND first before OR clause.

WILDCARDS (% / \_)is special character used to match parts of a value. Can only be used with STRINGS, cannot be used for non-text datatypes. Wildcards will not match NULLS. NULL represents no value in a column. Wildcards takes longer to run.

A table with text and images

Description automatically generated with medium confidence A close-up of a card

Description automatically generated

A black rectangular object with white text

Description automatically generated

SORTING with ORDER BY , data displayed appears in the order of the underlying tables. Sorting data helps keep information you want on top by particular columns(can’t see the whole of data in one view if you have a lot of data). SELECT something FROM database ORBER BY characteristic. ORDER BY can take multiple columns, with add a comma.. and sort by ORDER BY Clause **MUST ALWAYS** be the last statements in SQL Statement.

A computer screen shot of a black background

Description automatically generatedA black background with white text

Description automatically generated

MATH OPERATIONS :

1. + : Addition
2. - : Subtraction
3. \* : Multiplication
4. / : Division

**ORDER of Math Operations : Parentheses, Exponents, Multiplication, Division, Addition, Substraction.**

A screenshot of a computer

Description automatically generatedAggregate Functions to summarize our data (in math datatypes):

1. AVERAGE : to Averages a column of values
2. COUNT : Counts the number of values
3. MIN : finds the maximum value
4. MAX : finds the minimum value
5. SUM : Sums the column values.

IF DISTINCT (berbeda) is not specified, all is assumed. And cannot use DISTINCT on COUNT.

GROUPING DATA With SQL : HAVING statements (filter four groups).

SELECT something FROM database GROUP BY BY characteristic

Group By clauses can contain multiple columns. Every column in your SELECT Statement must be present in a GROUP BY Clause **except** for aggregated calculations. And NULL will be grouped together if your GROUP BY column contains NULLS.

WHERE doesn’t work for group, where filters on ROWS, that’s why we need to use HAVING clause to filter for groups.

A screen shot of a computer program

Description automatically generatedCount the order of customers, but we only want to see

The total order for the customers who have 2 orders.

WHERE filters **before** data is grouped

HAVING filters **after** data is grouped.

ROWS eliminated by the WHERE Clause will not be a included in the group

ORDER BY sorts data

GROUP BY does not sort data, only groups together.

1. What is the pay type for all the job codes that start with '03'? The code has been started for you, but you will need to program the fourth and fifth lines yourself before running the query.

#use WILDCARD

SELECT job\_code, pay\_type FROM salary\_range\_by\_job\_classification –table WHERE job\_code LIKE '03%'

1. What is the maximum biweekly high rate of pay (please include the dollar sign and decimal point in your answer)? The code has been started for you, but you will need to add onto the last line of code to get the correct answer.

SELECT  Max(Biweekly\_high\_Rate)  FROM salary\_range\_by\_job\_classification

1. Run a query to find the Effective Date (eff\_date) or Salary End Date (sal\_end\_date) for grade Q90H0? The code has been started for you, but you will need to program the third through the sixth lines yourself before running the query.

Select grade,eff\_date,sal\_end\_date FROM salary\_range\_by\_job\_classification WHERE grade LIKE '%90H0%'

1. Sort the Biweekly low rate in ascending order. There is no starter code, as you need to write and run the query on your own. Hint: there are 4 lines to run this query.

SELECT Biweekly\_Low\_Rate  FROM salary\_range\_by\_job\_classification ORDER BY Biweekly\_Low\_Rate ASC

1. Write and run a query, with no starter code to answer this question: **What Step are Job Codes 0110-0400?** Hint: there are 6 lines to run this query.

SELECT \* FROM salary\_range\_by\_job\_classification WHERE Job\_Code BETWEEN 0110 AND 0400

1. Write and run a query, with no starter code or hints to answer this question: **What is the Biweekly High Rate minus the Biweekly Low Rate for job Code 0170?**

SELECT Job\_code, Biweekly\_High\_Rate, Biweekly\_Low\_Rate, (Biweekly\_High\_Rate Biweekly\_Low\_Rate) as hasil FROM salary\_range\_by\_job\_classification WHERE Job\_Code = '0170'

1. Write and run a query, with no starter code or hints to answer this question: **What is the Extended Step for Pay Types M, H, and D?**

SELECT Job\_code, Extended\_Step , Pay\_Type FROM salary\_range\_by\_job\_classification WHERE Pay\_Type = 'D' OR  Pay\_Type = 'H' OR Pay\_Type = 'M'

1. Write and run a query, with no starter code or hints to answer this question: **What is the step for Union Code 990 and a Set ID of SFMTA or COMMN?**

SELECT Extended\_Step , SetID, Union\_code FROM salary\_range\_by\_job\_classification WHERE Union\_code= 990 AND (SetID = 'SFMTA'OR SetID = 'COMMN')

1. Select the true statement below. HAVING filters after the data is grouped.
2. A white text on a black background

   Description automatically generated
3. SELECT all that are true regarding wildcards (Select all that apply

Wildcards at the end of search patterns take longer to run

Wildcards take longer to run compared to a logical operator 5.

1. Select the statements below that ARE TRUE of the ORDER BY clause (Select all that apply).

It's only applied to the column names it directly precedes

Can take the name of one or more columns

1. Which of the following is true of GROUP BY clauses? (Select all that apply.)

NULLs will be grouped together if your Group By column contains NULLs

Every column in your select statement may/can be present in a group by clause, except for aggregated calculations

1. Question 10 Which is the correct order of occurrence in a SQL statement? select, from, where, group by, having

SQL for python : https://pypi.org/project/python-sql/

SQL for SQLite : <https://docs.python.org/3/library/sqlite3.html>

1. Find all the invoices whose total is between $5 and $15 dollars.

SELECT \* FROM INVOICES WHERE Total BETWEEN 5 AND 15

1. Find all the customers from the following States: RJ, DF, AB, BC, CA, WA, NY.

SELECT \* FROM Customers WHERE FirstName LIKE '%Jack%'

1. Find all the invoices for customer 56 and 58 where the total was between $1.00 and $5.00.

SELECT \* FROM Invoices WHERE (CustomerID = 56 OR CustomerID = 58) AND (Total BETWEEN 1 AND 5)

1. Find all the tracks whose name starts with 'All'.

SELECT \* FROM Tracks WHERE Name LIKE 'All%'

1. Find all the customer emails that start with "J" and are from gmail.com.

SELECT \* FROM Customers WHERE email LIKE 'J%'

1. Find all the invoices from the billing city Brasília, Edmonton, and Vancouver and sort in descending order by invoice ID.

SELECT \* FROM Invoices

WHERE BillingCity = 'Edmonton' OR BillingCity = 'Vancouver' OR BillingCity = 'Brasília' ORDER BY InvoiceID DESC

1. Find the albums with 12 or more tracks.

SELECT \* FROM TRACKS WHERE TrackId >=12

1. Find all the tracks that have a length of 5,000,000 milliseconds or more.

SELECT \* FROM TRACKS WHERE Milliseconds >= 5000000

1. Show the number of orders placed by each customer (hint: this is found in the invoices table) and sort the result by the number of orders in descending order.

SELECT \* FROM INVOICES ORDER BY CustomerID DESC

**CHAPTER II – Create, Read, Update, Delete (CRUD)**

DATATYPES tell a database management systems how to interpret the value of a column :

1. String : Storing data with mixed type of characters (Alphabet, Numeric and special).

* CHAR : the given length of character is predetermined (fix).
* VARCHAR : the given length of character is variable (not sure how many characters)

1. Numeric : Store data as numbers (Integer – whole number, and Decimal – Fraction value)
2. Date & Time
3. CREATE DATABASE cm\_devices;
4. mysql> use cm\_devices;
5. mysql> CREATE DATABASE cm\_devices;
6. mysql> CREATE TABLE devices( deviceID int, deviceName varchar(50) DEFAULT ‘MIAMI’, price decimal);
7. A black screen with white text

   Description automatically generatedmysql> show tables;

A black background with white text

Description automatically generated

1. mysql> show columns from devices;
2. mysql> CREATE TABLE stock(deviceID int NOT NULL, quantity int, totalCost decimal NOT NULL);

source :

1. <https://www.w3schools.com/sql/sql_datatypes.asp>
2. <https://learn.microsoft.com/en-us/sql/t-sql/data-types/decimal-and-numeric-transact-sql?view=sql-server-ver16&viewFallbackFrom=sql-server-ver16s>
3. <https://dev.mysql.com/doc/refman/8.0/en/numeric-types.html>

-CREATE DATABASE database\_name; --must be unique with 63 characters

-DROP DATABASE database\_name; --to remove a database

-CREATE TABLE table\_name (column1\_name DATATYPE…);

-ALTER TABLE table\_name ADD(column\_name DATA TYPE); --update column to existing table

-ALTER TABLE table\_name DROP(column\_name); --Drop or delete column from existing table

-ALTER TABLE table\_name MODIFY(column\_name NEW\_DATA TYPE); --modify column datatypefrom existing table

-INSERT INTO table\_name (column1\_name, column2\_name, column3\_name)

VALUES

(value11, value21, value31), --non numerical datatype value use with ‘’

(value12, value22, value32),

(value13, value23, value33);

-SELECT column\_name FROM table\_name;

-INSERT INTO target\_table (column\_name) SELECT column\_name FROM source\_table;

Source that provide extra knowledge on SQL DDL and DML commands :

1. <https://www.tutorialrepublic.com/sql-tutorial/sql-create-database-statement.php>
2. <https://www.tutorialspoint.com/sql/index.htm>
3. <https://www.javatpoint.com/sql-tutorial>

-UPDATE table\_name --SET clause specifies the location of the records within the table.

SET column1\_name = ‘column1’, column2\_name = ‘column2’ --non numerical VALUE datatype use with ‘’

WHERE column\_name = 'column’ ; -- WHERE is used to identify those records that fulfil a specified condition.

**DELETE NOT EQUAL WITH DROP**

-DELETE FROM table\_name WHERE column\_name = 'column’ ; --WHERE clause is the key for delete in rows.

=> **Delete command is to REMOVE a Record from a tables or deletes all records of data from a table,**

**without deleting the table.**

**=> TRUNCATE command is SQL statement deletes the data inside the customers table, but not the table itself.**

**=> DROP command is used to drop an existing database or a table in a database.**

**=> A truncate SQL statement is used to remove all rows (complete data) from a table. It is similar to the DELETE statement with no WHERE clause.**

1. The table name should be written after the TABLE keyword (CREATE TABLE staff (name VARCHAR(100), address VARCHAR(100));
2. The INSERT INTO command is used to insert new records in a table – INSERT INTO table\_name VALUE (value1)
3. UPDATE staff; --table name
4. The UPDATE command is used to modify data in the database.
5. UPDATE table\_name SET column\_name = ‘column’ WHERE column\_name = ‘column’
6. UPDATE staff SET email = 'Karl@email.com' WHERE name = 'Karl';

SOURCE to continue to explore database engineering :

1. <https://www.tutorialspoint.com/sql/index.htm>
2. <https://www.w3schools.com/sql/>
3. <https://www.javatpoint.com/sql-tutorial>