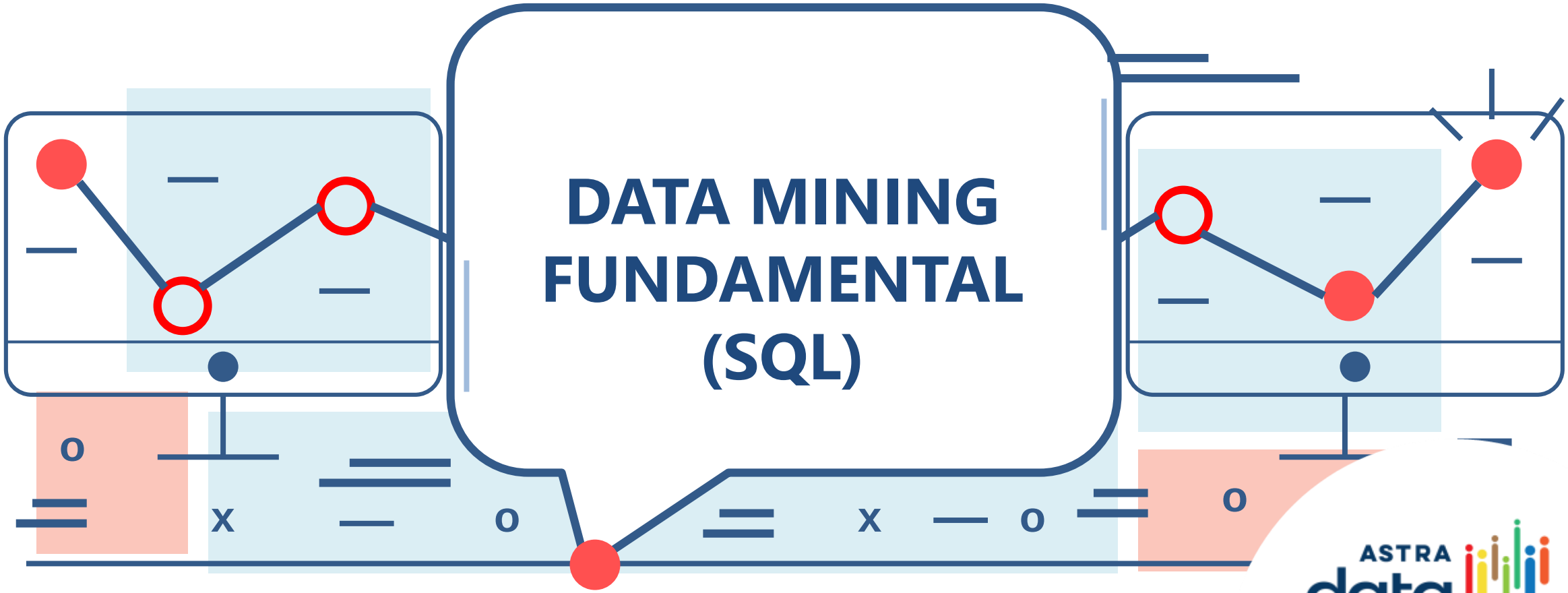
ASTRA

OFFICIAL PRESTIGE PARTNER

Jakarta Palembang 2018

satu Indonesia
Semangat Astra Terpadu Untuk Indonesia



DATA MINING FUNDAMENTAL (SQL)

iykra

Training Series: 120.051.11.111.29

ASTRA data scientist bootcamp

Yunan Putranto

Starting professional career in Oil and Gas Industry for a decade.
Extending expertise in Data Technology and its application focusing on but not limited to Big Data, Data Mining and Machine Learning.
Strong believer that education is the top powerful life-changer, and that data driven decision is most of the time the best decision.

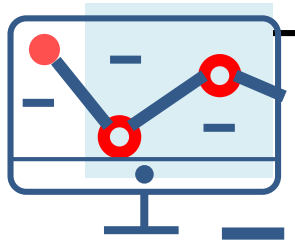
EDU

WORK

INTEREST

Email : yunanto.putranto@gmail.com, yunanto@datalabs.id
LinkedIn : https://www.linkedin.com/in/yunan-putranto-b045943b/

ASTRA data scientist bootcamp



Agenda

Session 0: Brief review of past topic

Session 1: PostgreSQL & DBeaver Installation, Data Loading

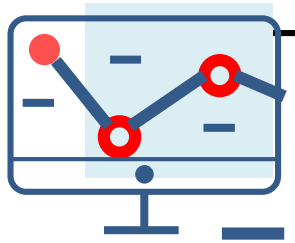
Session 2: Query Part1 – Basic level, Exercise

Break

Session 3: Query Part2 – Intermediate level, Exercise

Session 4: Assignment, Recap of the day

ASTRA data scientist bootcamp



Objective

Understand data mining fundamental

Handy with SQL command

Able to extract information from data in database

ASTRA data scientist bootcamp

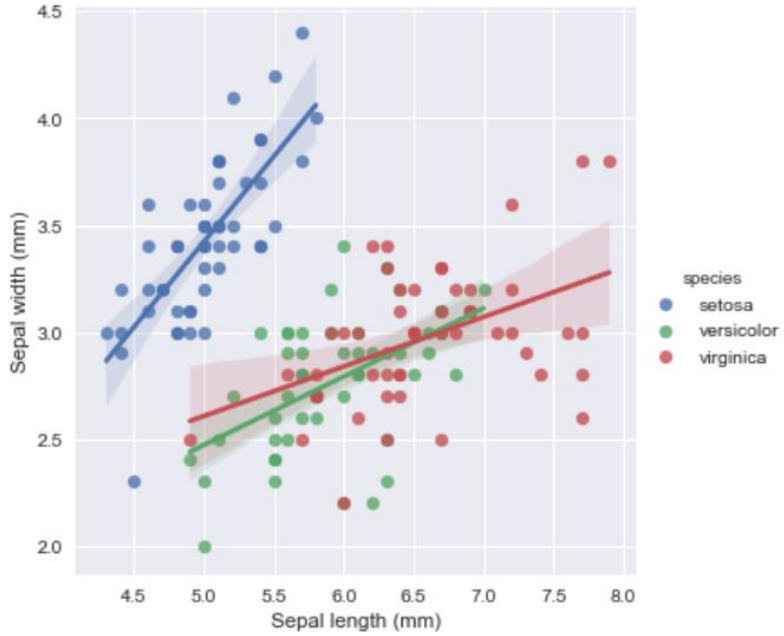
Session 0

Recap from past topic
‘Calculus’

Linear Function

- **Single Variable:** Input is of degree one or zero.
  - $f(x) = x$
  - $g(x) = 5x + 2$

- **Multi-Variables:** Inputs are of degree one or zero and no multiples of variables.
  - $f(x, y) = x + y$
  - $g(x, y) = 5x + 2y + 2$



Linear Regression

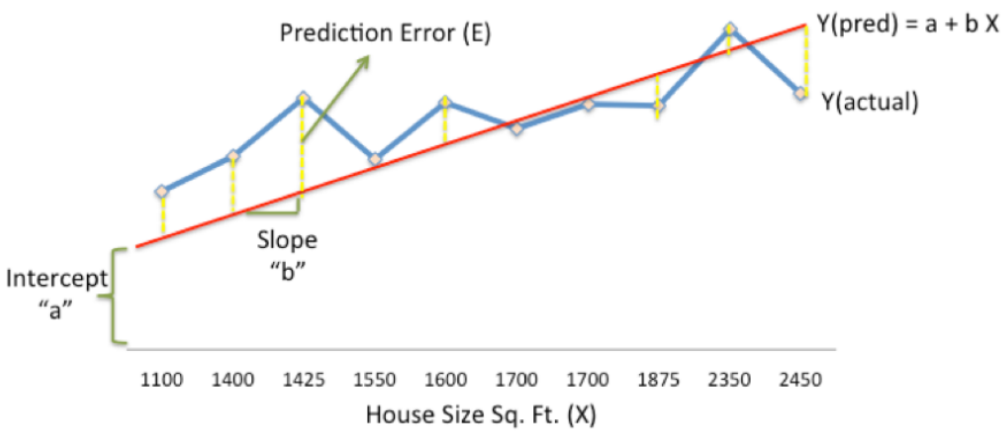
"Interesting things in the world are nonlinear (stock price, weather, etc). However human intuition is bad for nonlinearity, in the other hand good with linearity. That's why linear regression is very popular and useful"

ASTRA data scientist bootcamp

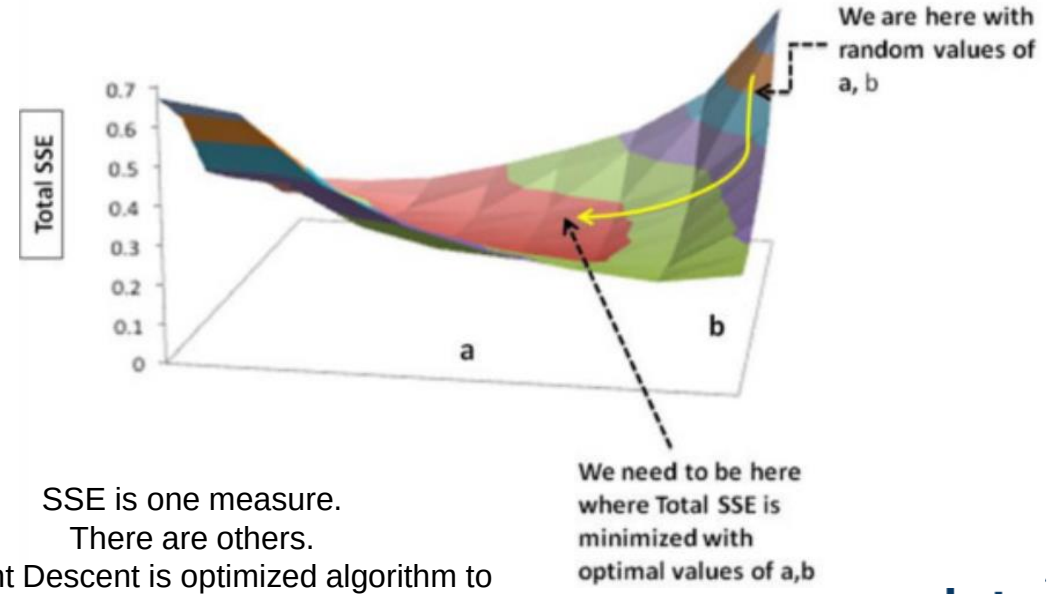
Derivative

- Given $f(x,y) = x^2 + y^2 + 2xy$
- Need to determined upon which variable we want to do our derivation

$$\frac{\partial f}{\partial x} = 2x + 0 + 2y \qquad \frac{\partial f}{\partial y} = 0 + 2y + 2x$$



Find a and b with minimum SSE (Sum of Square Error)

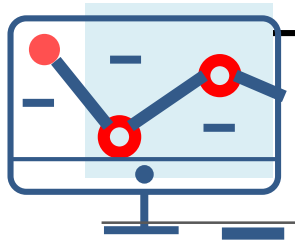


SSE is one measure.
There are others.
Gradient Descent is optimized algorithm to find a and b with minimum error

data scientist bootcamp

Session 1

- ✓ Data Mining
- ✓ SQL
- ✓ PostgreSQL Installation
- ✓ DBeaver Installation
- ✓ Data Loading
- ✓ DDL



Data Mining

da·ta min·ing

noun COMPUTING

noun: **data mining**; noun: **datamining**

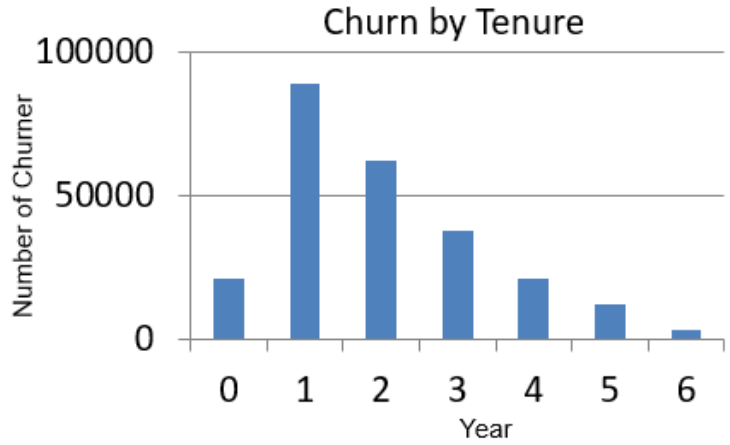
the practice of examining large databases in order to generate new information.

- Process of discovering **patterns** in **large dataset** involving methods at the intersection of machine learning, statistics and database systems (source)
- Process to turn **raw data** to **useful information** (source)
- Process of discovering interesting and **useful patterns** and relationship in **large volumes of data**. Also known as knowledge discovery. (source)

ASTRA data scientist bootcamp

Data Mining Example

- **Automobile**
  Proactive Maintenance Event
- **Service Provider**
  Churn Prediction
- **E-commerce**
  ‘People who view this also view this’
- **Supermarket**
  ‘Predict shopper likely to be pregnant’
- **More and more areas**



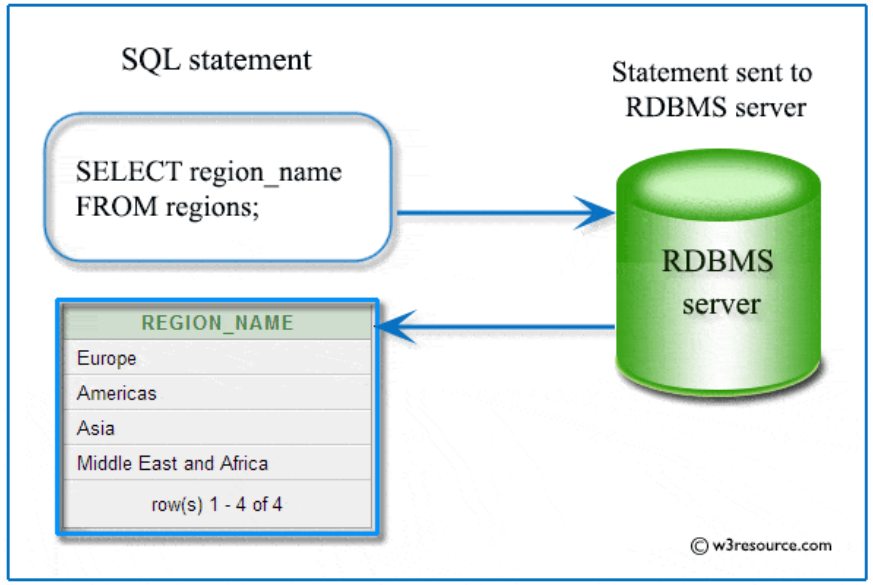
ASTRA data scientist bootcamp

SQL

- **Structured Query Language,** query language to interact with data in RDBMS (Relational Data Base Management System). (Created in 1970).

Why do we learn SQL?

- Data mining
- Data manipulation
- Combine data from multiple sources
- Manage large pool of data
- High demand
- And many more



SQL

ASTRA data scientist bootcamp

SQL command

3 types of SQL command

DDL (Data Definition Language)

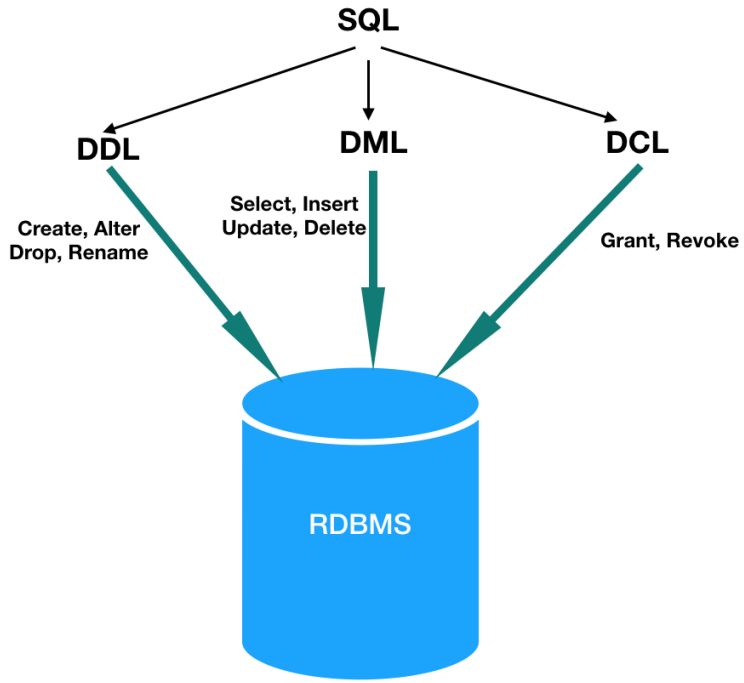
- Changes structure of table.
- Create, alter and delete table.
- Auto-committed, save all the changes permanently.

DML (Data Manipulation Language)

- Manipulates data in table.
- Insert, update, select etc

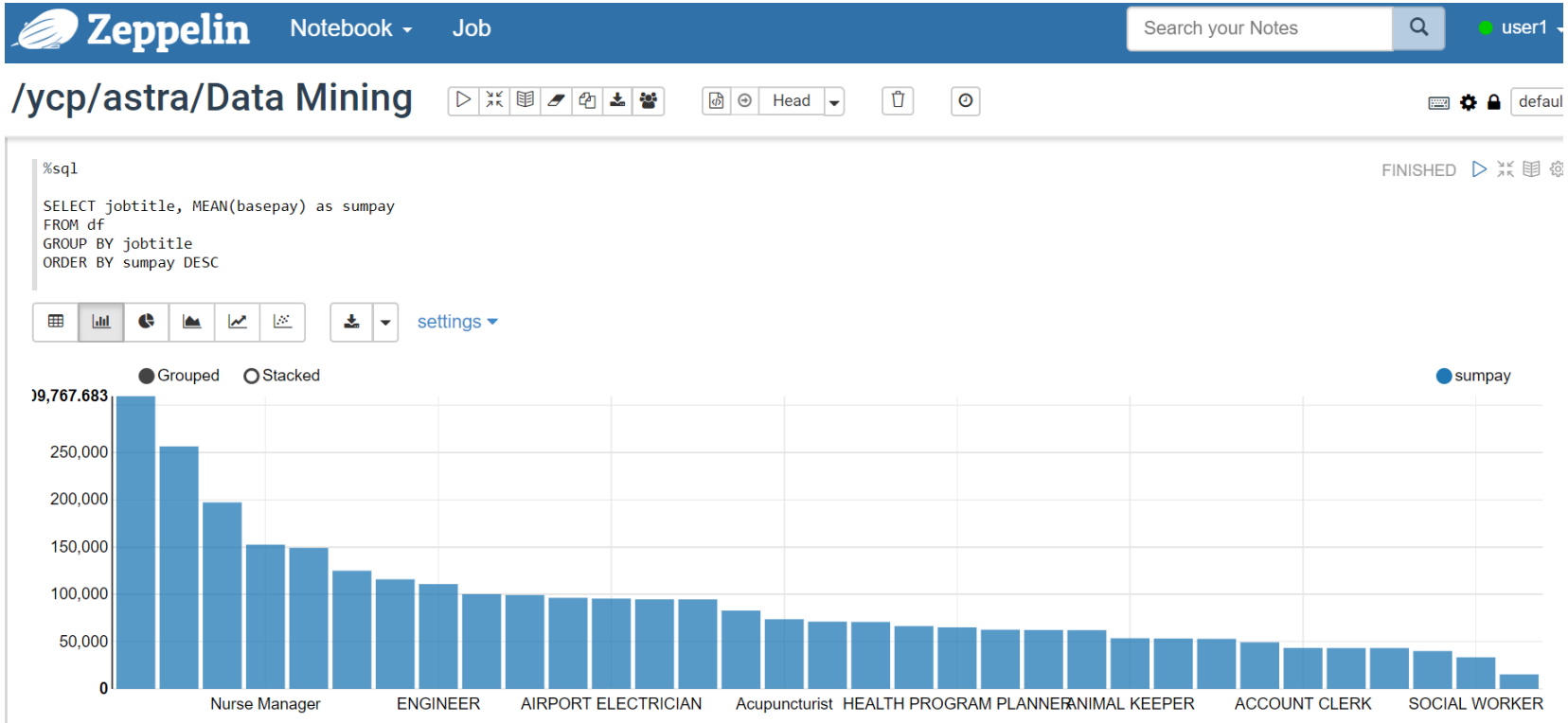
DCL (Data Control Language)

- Grand and take back authority from any database user.



ASTRA data scientist bootcamp

SQL with Zeppelin Demo



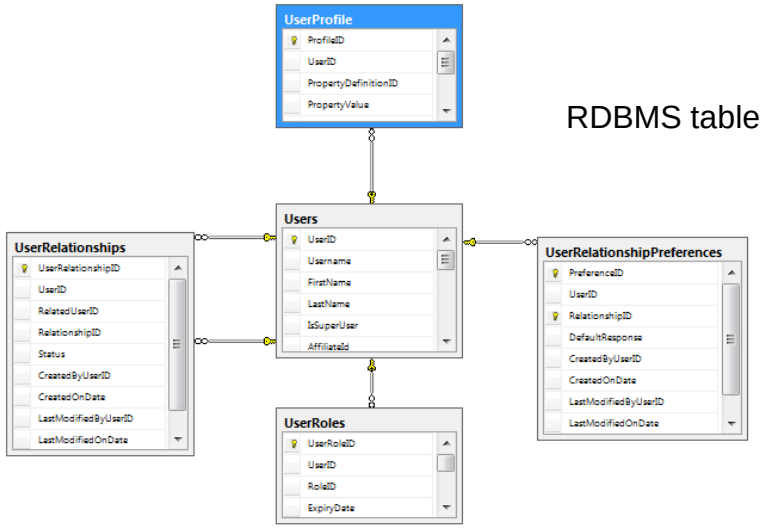
[Zeppelin Notebook Link]

ASTRA data scientist bootcamp

PostgreSQL & DBeaver

PostgreSQL (Postgres)

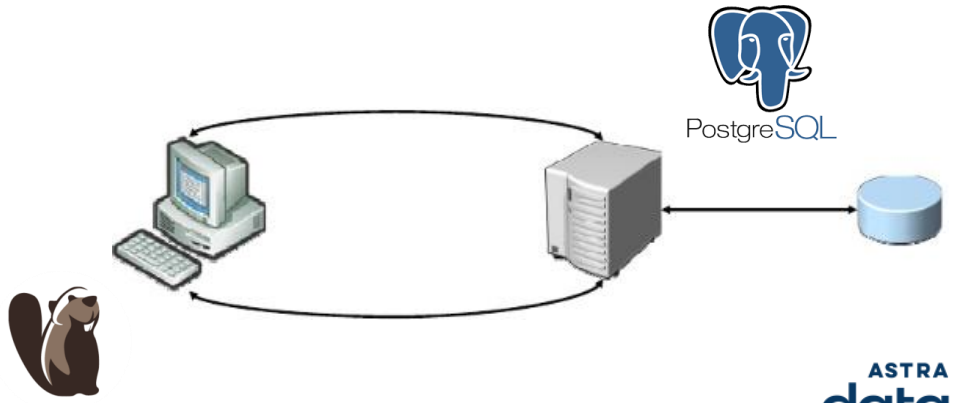
- RDBMS
- Database server
- Open Source



RDBMS table relationship diagram

DBeaver

- SQL client
- Database administration tool
- Open source (under Apache License)

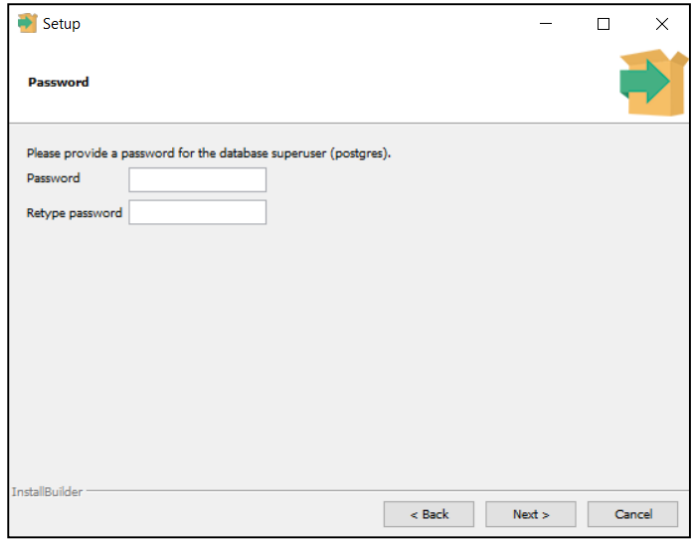


ASTRA data scientist bootcamp

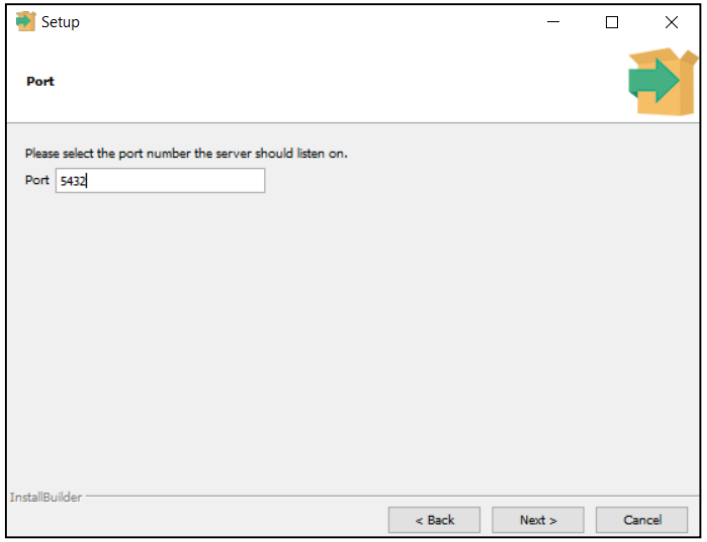
PostgreSQL Installation

- Run this installer: [postgresql-9.5.13-1-windows-x64]

Keep all default.
Hit Next, Next, Next.



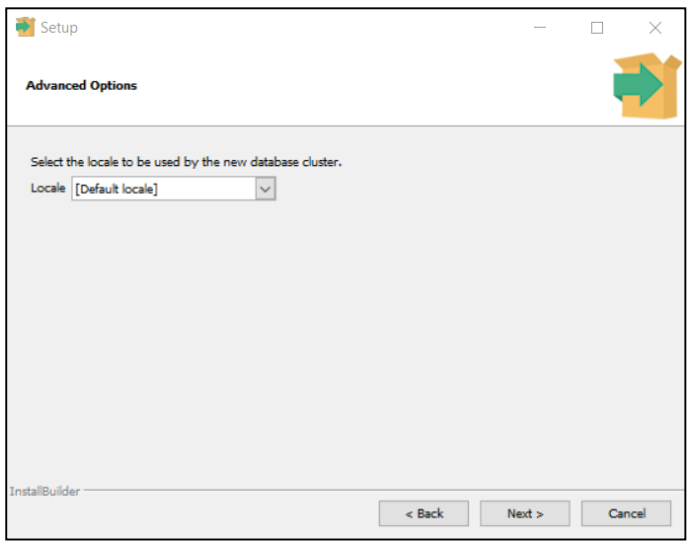
Type your own password.



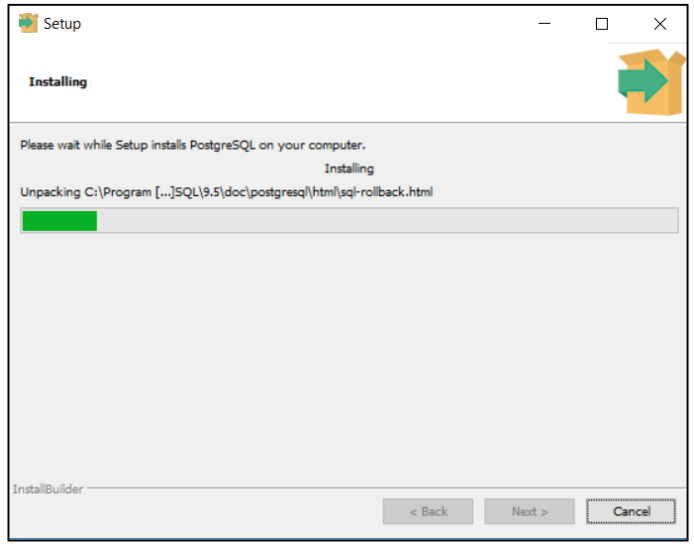
Keep default Port.

ASTRA data scientist bootcamp

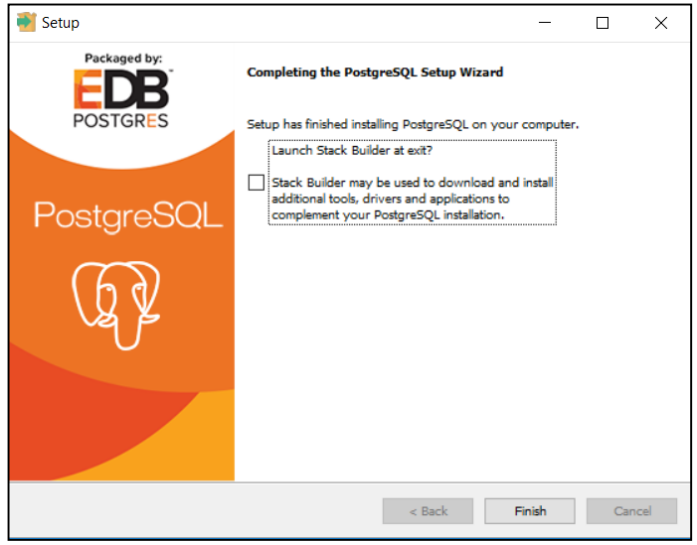
PostgreSQL Installation



Keep all default.
Hit Next, Next, Next.



Wait until installation completed.
Hit Next.

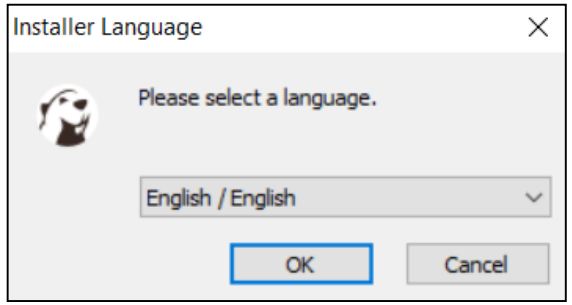


Deselect the Stack Builder.
Finish !

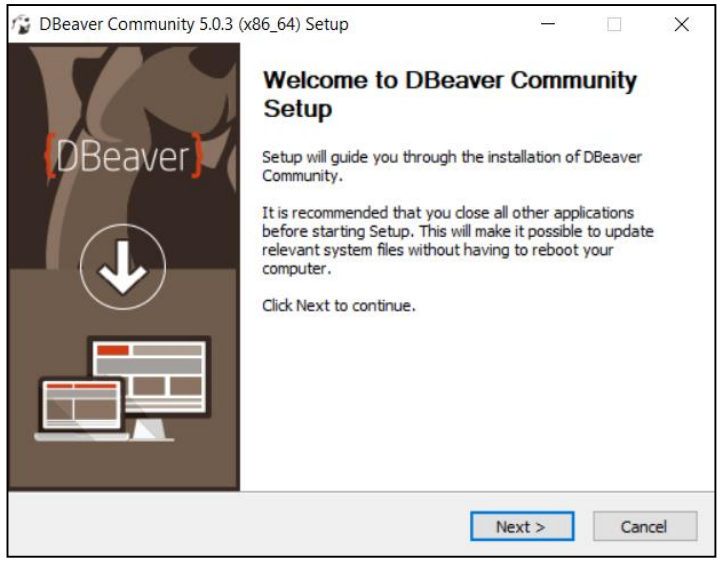
ASTRA data scientist bootcamp

DBeaver Installation

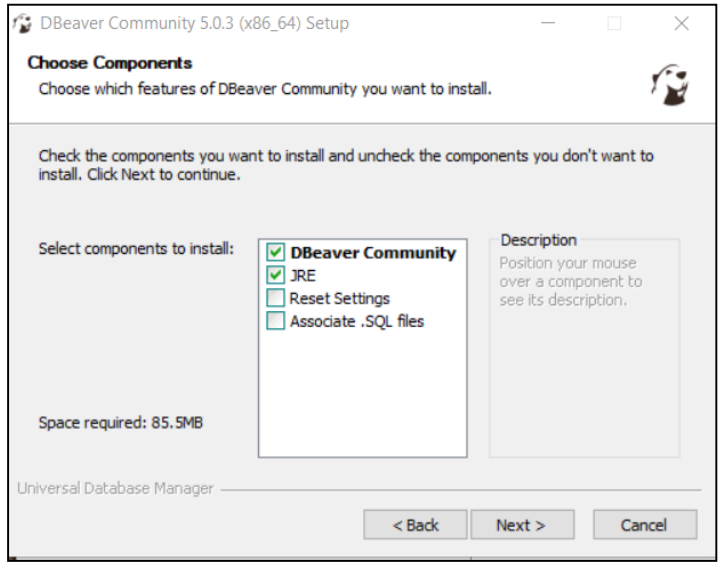
- Run this installer: dbeaver-ce-5.0.3-x86_64-setup



Keep default (English)
Ok.



Next / I Agree / Next.

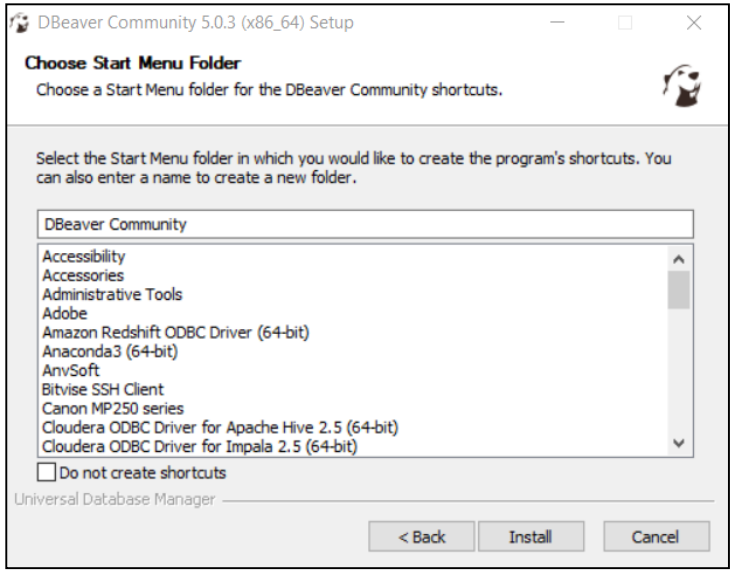


Keep default. Next.

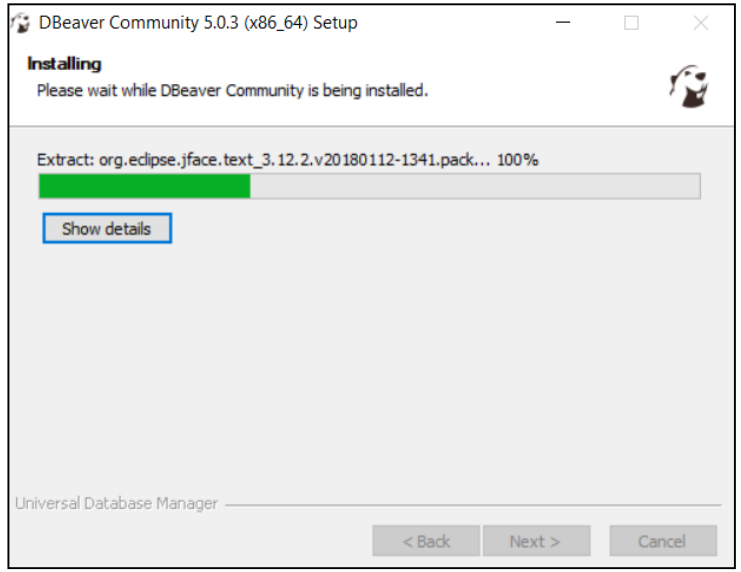
ASTRA data scientist bootcamp

DBeaver Installation

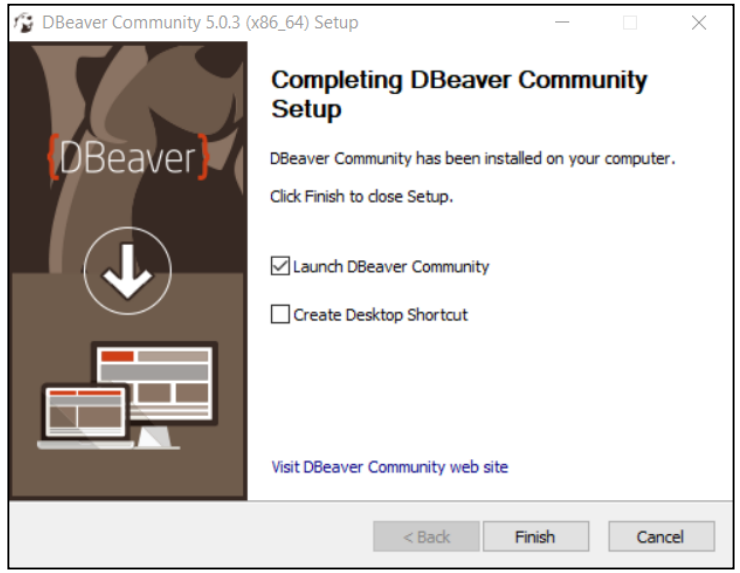
- Run this installer: dbeaver-ce-5.0.3-x86_64-setup



Keep default. Install.



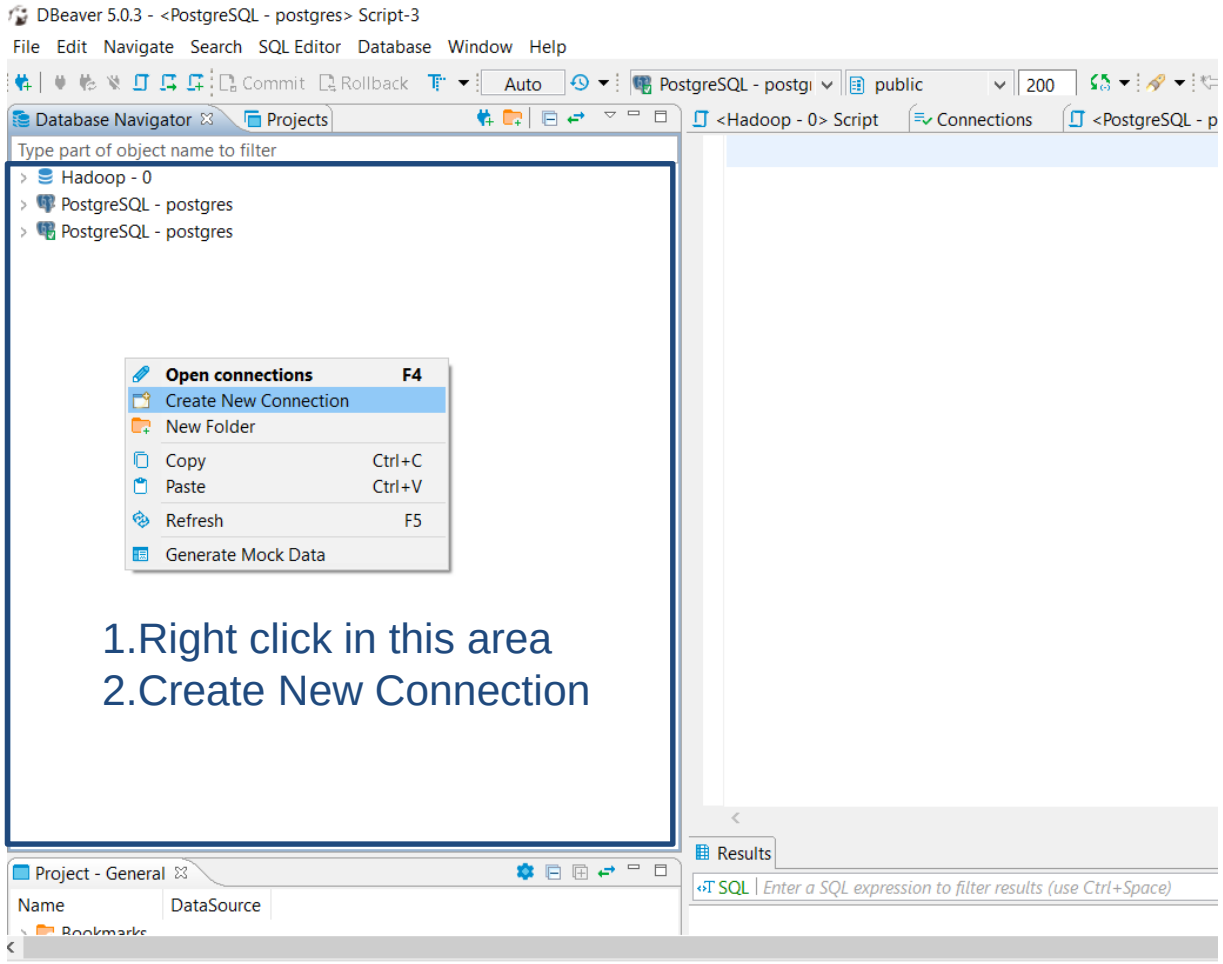
Wait until installation completed.
Hit Next.

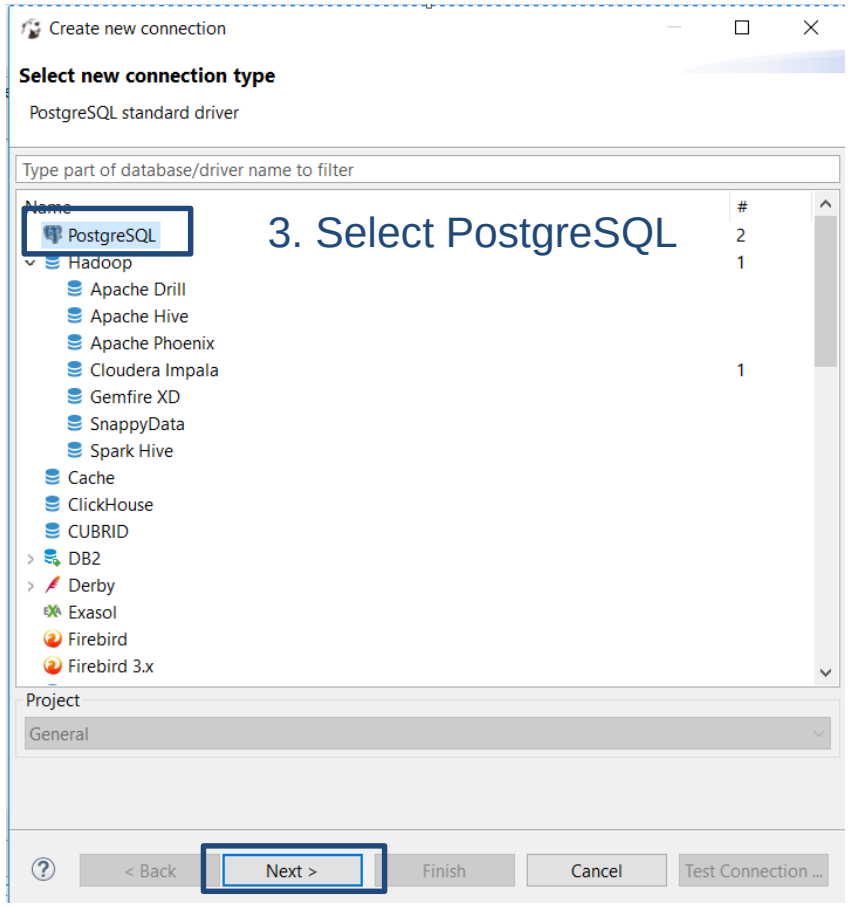


Finish !

ASTRA data scientist bootcamp

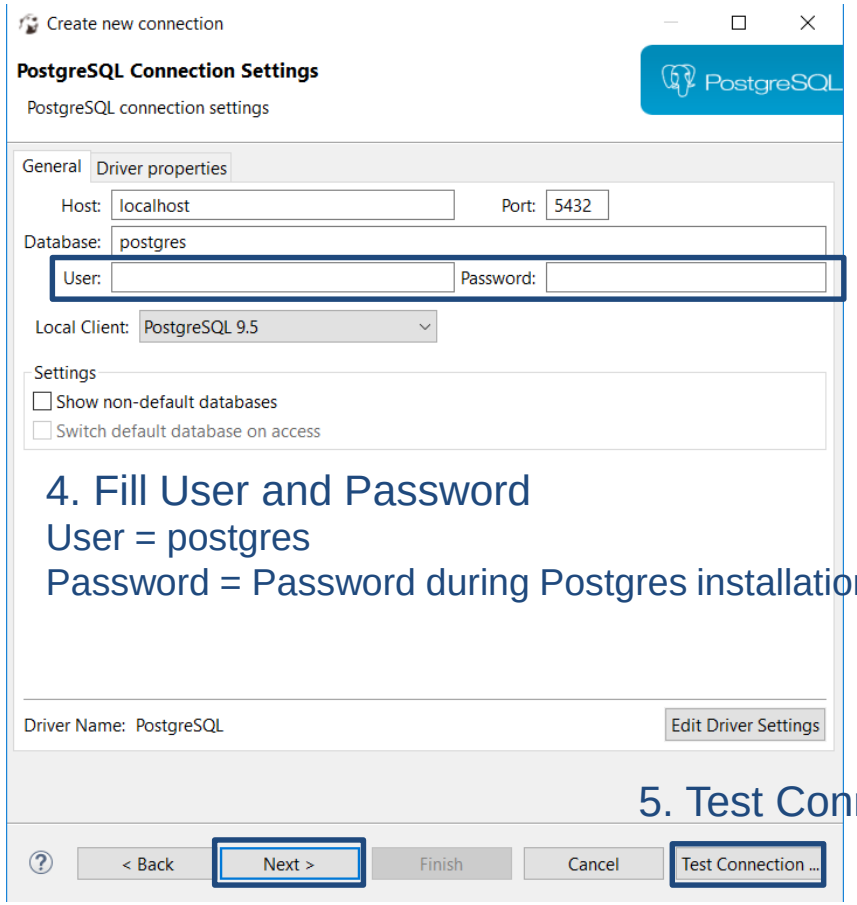
Connect to Postgres from DBeaver





ASTRA data scientist bootcamp

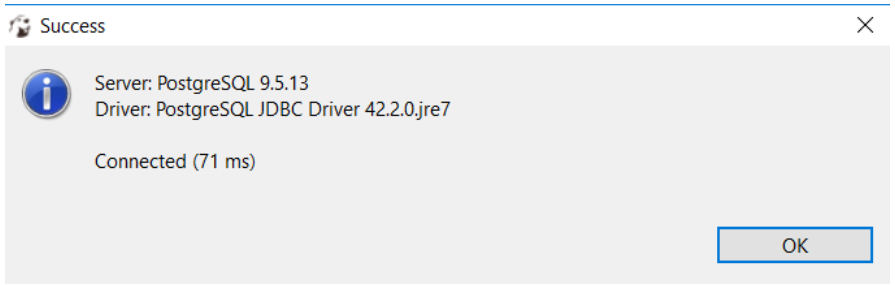
Connect to Postgres from DBeaver



4. Fill User and Password
User = postgres
Password = Password during Postgres installation

5. Test Connection

6. Next, Next until Finish.

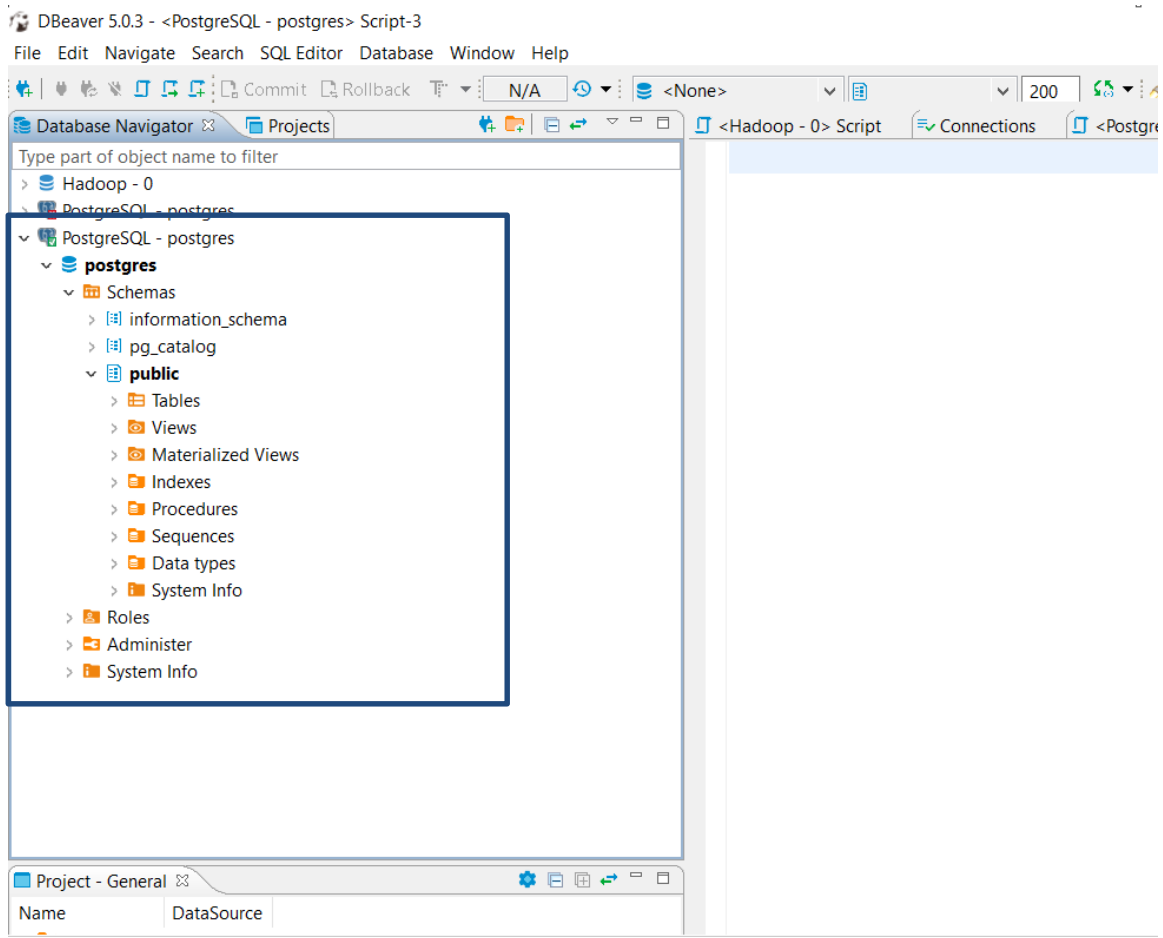


Test Connection Successful

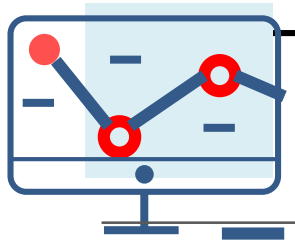
ASTRA data scientist bootcamp

Connection established!

This is our database.
Table will be created under ‘public’.



ASTRA data scientist bootcamp



Create Tables, Load Data

- **Create table in postgres**

Data for exercise: Salaries data

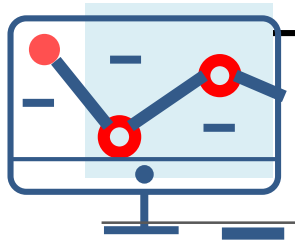
Field: Id, EmployeeName, JobTitle, BasePay, OvertimePay, OtherPay, TotalPay, Years

- **Load data**

Load .csv file into existing table in postgres

-- Practical --

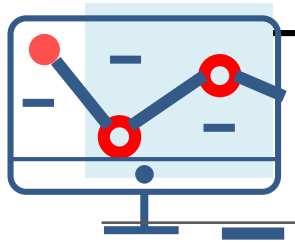
ASTRA data scientist bootcamp



DDL - CREATE Table

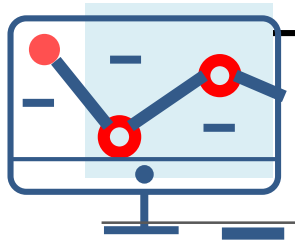
```
CREATE TABLE SALARIES
(
Id SERIAL NOT NULL,
EmployeeName VARCHAR(50),
JobTitle VARCHAR(30),
BasePay DECIMAL(8,2),
OvertimePay DECIMAL(8,2),
OtherPay DECIMAL(8,2),
TotalPay DECIMAL(8,2),
PRIMARY KEY (Id)
);
```

ASTRA data scientist bootcamp



DDL - ALTER Table

- **Add column** to existing table
- **Rename** any existing column
- **Change datatype** of any column or to modify its size.
- **Drop column** from the table.



Example ALTER Table

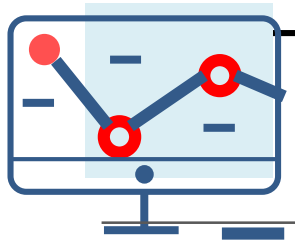
- Add new column **tahun** in **salaries** table using ALTER command:

```
ALTER TABLE salaries ADD(
 Tahun VARCHAR(4)
);
```

- Modify column **TotalPay** in **salaries** table using ALTER command:

```
ALTER TABLE salaries MODIFY(
TotalPay DECIMAL(10,2)
);
```

ASTRA data scientist bootcamp



Example ALTER Table

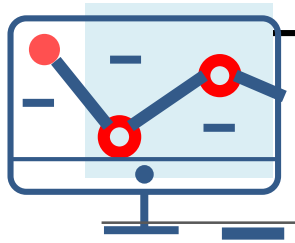
- Rename column **Tahun** in **salaries** table to **Years** using ALTER command:

```
ALTER TABLE salaries RENAME
Tahun to Years;
```

- Drop column **Years** in **salaries** using ALTER command:

```
ALTER TABLE salaries DROP(
Years
);
```

ASTRA data scientist bootcamp



DDL - RENAME Table

- **RENAME** command is used to set a new name for any existing table.

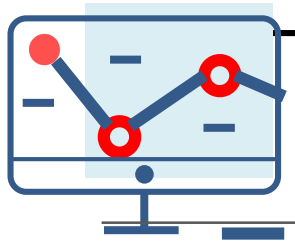
Syntax:

RENAME TABLE old_table_name **to** new_table_name**;**

Example:

RENAME TABLE salaries **to** wage**;**

ASTRA data scientist bootcamp



DDL - TRUNCATE & DROP

- **TRUNCATE** command removes all records from a table but not deleting table structure

Syntax:

TRUNCATE TABLE table_name**;**

Example:

TRUNCATE TABLE salaries**;**

- **DROP** command completely removes the table from database.

Syntax:

DROP TABLE table_name**;**

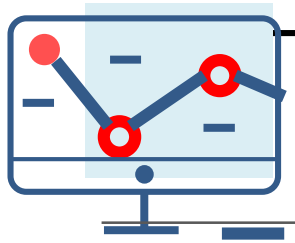
Example:

DROP TABLE salaries**;**

ASTRA data scientist bootcamp

Session 2

- ✓ Query Part-1
- ✓ SELECT, WHERE, ORDER, UPDATE, INSERT, DELETE



SELECT Statement

Select specify column:

SELECT s_id, name, age FROM student;

To select all columns and all rows, use * as abbreviation for all columns:

SELECT * FROM student;

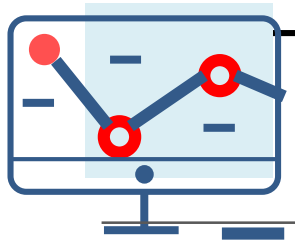
With clausa WHERE:

SELECT * FROM student WHERE name = 'Abhi';

With DISTICNT to retrieve unique values from the table:

SELECT DISTINCT salary from Emp;

ASTRA data scientist bootcamp



SELECT Statement (cont)

With LIKE clause:

SELECT * FROM Student WHERE s_name LIKE 'A%';

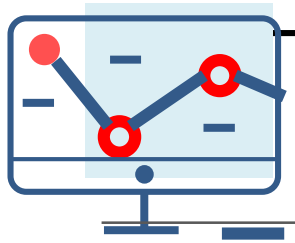
To name column, use AS clause:

SELECT s_id, name, age AS umur FROM student;

With AND & OR operator for comparison:

SELECT * FROM Emp WHERE salary < 10000 AND age > 25;

SELECT * FROM Emp WHERE salary < 10000 OR age > 25;



SELECT Statement (cont)

With BETWEEN to search range condition:

SELECT * FROM Emp WHERE salary BETWEEN 10000 AND 30000;

With IN and NOT IN to filter row:

SELECT * FROM staff WHERE position IN (‘Manager’, ‘Supervisor’);

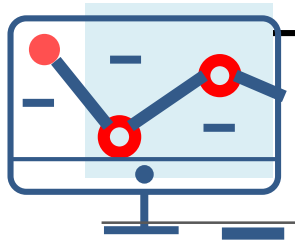
SELECT * FROM staff WHERE position NOT IN (‘Manager’, ‘Supervisor’);

With NULL and NOT NULL condition:

SELECT * FROM Emp WHERE age = 25 AND salary is NULL;

SELECT * FROM Emp WHERE age > 25 and salary is NOT NULL;

ASTRA data scientist bootcamp



SELECT Statement (cont)

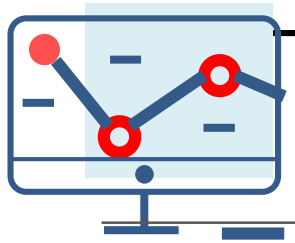
With ORDER BY clause for arranging data in sorted order based on one or more colums:

SELECT * FROM Emp ORDER BY salary;

SELECT * FROM Emp ORDER BY salary DESC;

With GROUP BY clause for grouping data based on one or more columns:

SELECT name, salary FROM Emp WHERE age > 25 GROUP BY salary;



SELECT Statement (cont)

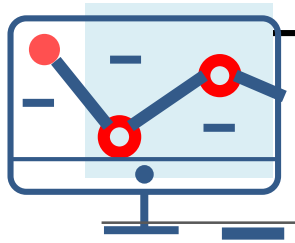
Consider the following **Sale** table.

oid	order_name	previous_balance	customer
11	ord1	2000	Alex
12	ord2	1000	Adam
13	ord3	2000	Abhi
14	ord4	1000	Adam
15	ord5	2000	Alex

With HAVING clause to give more precise condition:

SELECT * FROM sale GROUP BY customer

HAVING sum(previous_balance) > 3000;



UPDATE Statement

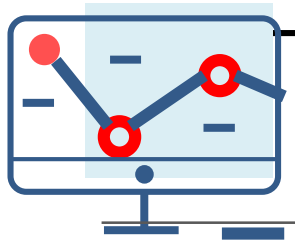
UPDATE command is used to update any record of data in table.

Syntax:

UPDATE table_name SET column_name = new_value WHERE some_condition;

Example:

UPDATE student SET name='Abhi', age=17 where s_id=103;



INSERT Statement

INSERT command is used to insert data into table.

Syntax for all columns in table:

INSERT INTO table_name VALUES (data1,data2,…)

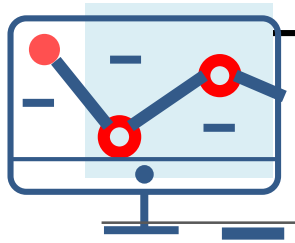
Syntax for specify columns:

INSERT INTO table_name (column1,column2,…) VALUES(data1,data2,…)

Example:

INSERT INTO student VALUES (101, 'Adam', 15);

INSERT INTO student(id, name) values(102, 'Alex');



DELETE Statement

DELETE command is used to remove data from table.

Syntax delete all record in table:

DELETE FROM table_name;

Syntax delete for particular record in table using WHERE clause:

DELETE FROM table_name WHERE *conditions;*

Example remove all records:

DELETE FROM student;

Example remove with condition:

DELETE FROM student WHERE s_id = 103;

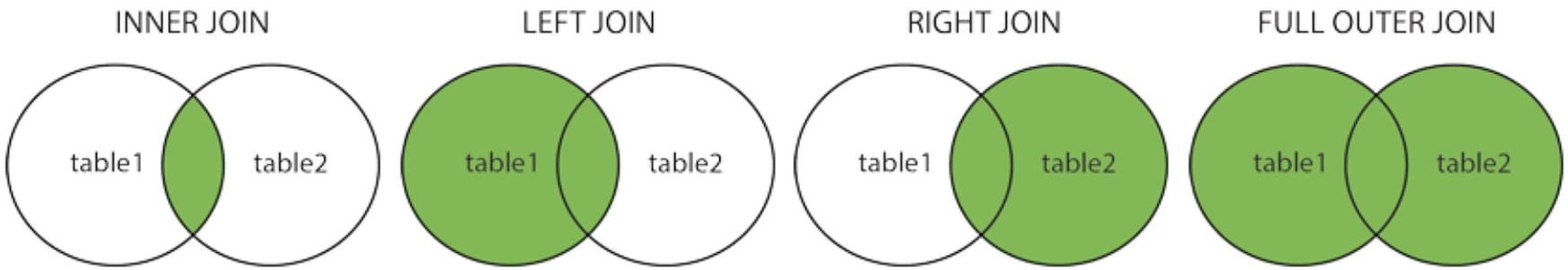
ASTRA data scientist bootcamp

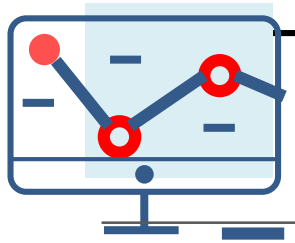
Session 3

- ✓ Query Part-2
- ✓ AGGREGATION, JOIN, SUBQUERY, WINDOW FUNCTION

JOIN

A JOIN clause is used to combine rows from two or more tables, based on related columns between them.





INNER JOIN

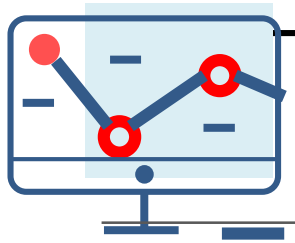
INNER JOIN selects records that have matching values in both tables.

Syntax:

```
SELECT column_name(s)
FROM table1
INNER JOIN table2 ON table1.column_name = table2.column_name;
```

Example:

```
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
```



LEFT JOIN

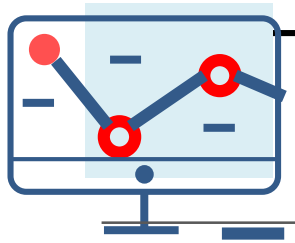
LEFT JOIN return all records from the left table (table1) and the matched records from the right table (table 2)

Syntax:

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2 ON table1.column_name = table2.column_name;
```

Example:

```
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders
LEFT JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
```



RIGHT JOIN

RIGHT JOIN return all records from the right table (table 2) and the matched records from the left table (table1)

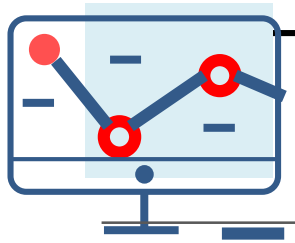
Syntax:

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2 ON table1.column_name = table2.column_name;
```

Example:

```
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders
RIGHT JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
```

ASTRA data scientist bootcamp



FULL JOIN

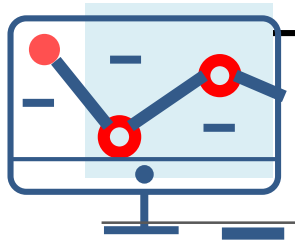
FULL JOIN return all records when there is match in either left table (table1) or right table (table2)

Syntax:

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2 ON table1.column_name = table2.column_name;
```

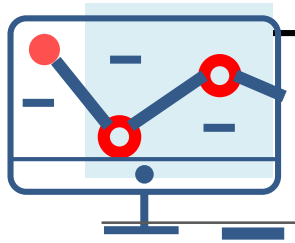
Example:

```
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders
FULL OUTER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
```



Aggregates Function

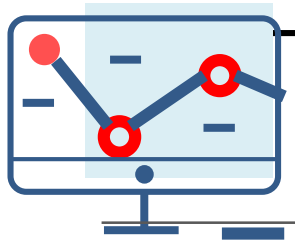
- **COUNT()** returns number of values in selected column.
- **SUM()** returns sum of values in selected column.
- **MAX()** returns largest of values in specified column.
- **MIN()** returns smallest of values in specified column.
- **AVG()** returns average of values in specified column.



Aggregates Function

Rules in aggregates function:

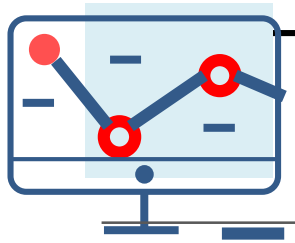
- Each operates on a single columns of a table return a single value.
- COUNT(), MIN() and MAX() apply to numeric and non-numeric variables. While SUM() and AVG() must be used on numeric variables only.
- Aprt from COUNT(*), each function eliminates nulls first and operates only on remaining non-null values.



Aggregates Function (cont.)

Rules in aggregates function:

- COUNT(*) count all rows of table, whether null values or duplicates occur.
- Can use DISTINCT before column name to eliminate duplictes.
- DISTINCT has no effect with MIN()/MAX(), but may have with SUM()/AVG().
- Aggregate function must have GROUP BY clause.
- Aggregate function can be used only in SELECT list and in HAVING clause.



Aggregates Function (count)

- COUNT()

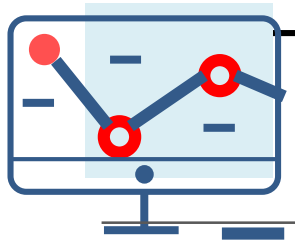
  Example: How many employee in 2011 from table Salaries?

  SELECT COUNT(*) AS NumEmployee
  FROM salaries
  WHERE years = '2011';

- COUNT () & SUM()

  Example: Count number of managers and sum of their salaries.

  SELECT COUNT(staffNo) AS Mycount, SUM(salary) AS mysum
  FROM staff
  WHERE position = 'Manager';

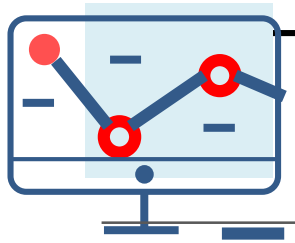


Aggregates Function (count)

- MIN(), MAX(), AVG()

  Example: Find minimum, maximum and average staff salary.

  SELECT MIN(salary) AS minsal, MAX(salary) AS maxsal,
  AVG(salary) AS avesal
  FROM staff;

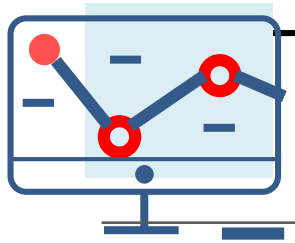


Aggregates Function (count)

Aggregate function also can combine with GROUP BY clause.

Example: Find number of staff in each branch and their total salaries.

```
SELECT branchNo, COUNT(staffId) AS mycount, SUM(salary) AS mysum
FROM staff
GROUP BY branchNo
Order BY branchNo;
```

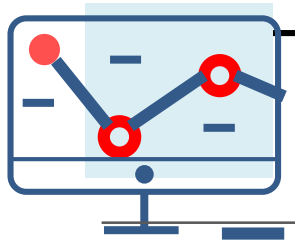


Aggregates Function (count)

Aggregate function also can combine with HAVING clause where have specify condition.

Example: For each branch with more than 1 member of staff, find number of staff in each branch and sum their salaries.

```
SELECT branchNo, COUNT(staffId) AS mycount, SUM(salary) AS mysum
FROM staff
GROUP BY branchNo
HAVING COUNT(staffId) > 1
Order BY branchNo;
```

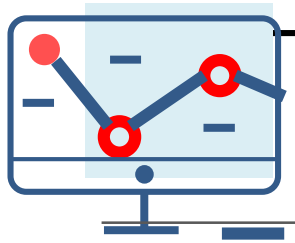


Window Function

A Window function performs a calculation across a set of table rows that are somehow related to the current row. This is comparable to the type of calculation that can be done with an aggregate function.

SELECT <column names or functions> OVER (<PARTITION BY, ORDER BY>)
FROM <tablenames>

ASTRA data scientist bootcamp



Window Function (cont)

Example:
Calculate average salary in each department to compare each employee's salary.

SELECT depname, empno, salary, avg(salary) OVER (PARTITION BY depname) FROM empsalary;

```
  depname  | empno | salary |          avg
-----------+-------+--------+-----------------------
 develop   |    11 |   5200 | 5020.0000000000000000
 develop   |     7 |   4200 | 5020.0000000000000000
 develop   |     9 |   4500 | 5020.0000000000000000
 develop   |     8 |   6000 | 5020.0000000000000000
 develop   |    10 |   5200 | 5020.0000000000000000
 personnel |     5 |   3500 | 3700.0000000000000000
 personnel |     2 |   3900 | 3700.0000000000000000
 sales     |     3 |   4800 | 4866.6666666666666667
 sales     |     1 |   5000 | 4866.6666666666666667
 sales     |     4 |   4800 | 4866.6666666666666667
(10 rows)
```

ASTRA data scientist bootcamp

Window Function (cont)

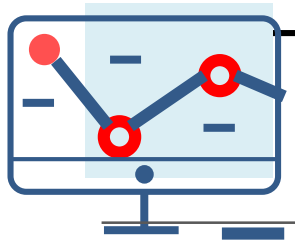
A Window function can control the order in which row are processed using **ORDER BY within OVER()**.

Example:
Calculate rank for each employee with order by salary descending in each department.

SELECT depname, empno, salary, rank() OVER (PARTITION BY depname ORDER BY salary DESC) FROM empsalary;

depname	empno	salary	rank
develop	8	6000	1
develop	10	5200	2
develop	11	5200	2
develop	9	4500	4
develop	7	4200	5
personnel	2	3900	1
personnel	5	3500	2
sales	1	5000	1
sales	4	4800	2
sales	3	4800	2

(10 rows)



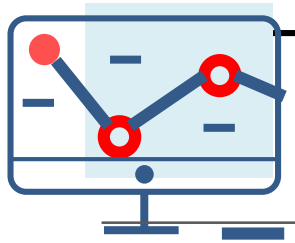
Subqueries

- Some SQL statement may have a SELECT embedded within them.
- A subselect can be used in WHERE and HAVING clause, INSERT, UPDATE, DELETE statement or inside another subquery.

Example: List staff who work in branch at ‘163 Maint st’.

```
SELEC TstaffNo, fName, lName, position
FROM Staff
WHERE branchNo=
        (SELECT branchNo
          FROM Branch
          WHERE street=‘163MainSt’);
```

staffNo	fName	lName	position
SG37	Ann	Beech	Assistant
SG14	David	Ford	Supervisor
SG5	Susan	Brand	Manager



Subqueries (cont)

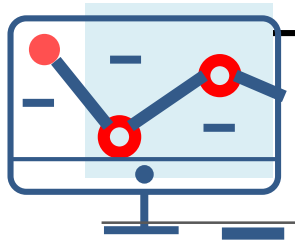
Subqueries can be combined with window function when there's a need to filter or group rows after the window calculation performed.

Example:
This query shows row from the inner query which having rank less than 3.

```
SELECT depname, empno, salary, enroll_date
FROM
  (SELECT depname, empno, salary, enroll_date,
          rank() OVER (PARTITION BY depname ORDER BY salary DESC, empno) AS pos
     FROM empsalary
  ) AS ss
WHERE pos < 3;
```

Session 4

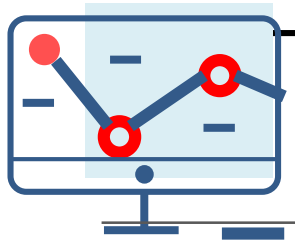
- ✓ Assignment
- ✓ Recap of the day



Assignment

- 10 questions of SQL query covering all material of the day
- Duration max 45 minutes
- Review

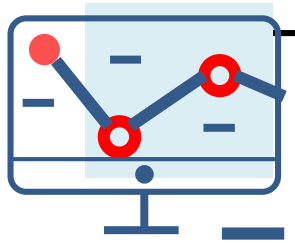
ASTRA data scientist bootcamp



It’s a wrap

- **Data Mining** definition
- **SQL Command: DDL** (Create, Alter, Delete)**, DML** (Insert, update, select)**, DCL**
- **Establish connection** from Dbeaver to Postgres
- **Create table** in database
- **Perform DDL command**
- **Load data** (.csv) into existing table in database
- **Perform DML command**
- **Basic Query** SELECT, WHERE, ORDER, INSERT, UPDATE, DELETE
- **More advance Query** AGGREGATE, JOIN, SUBQUERY, WINDOW FUNCTION

ASTRA data scientist bootcamp



Thank you

IYKRA

Ariobimo Sentral Kuningan, Block 71 Lantai 8.

Jl. HR Rasuna Said Blok X-2, Kav. 6, Kuningan

Jakarta 12950