

# SQL DML – simple SELECT

- Intro SQL DML
- Basic SQL query
- Eliminating duplicate rows – DISTINCT
- Ordering (ORDER BY)
- Exercises – SQL simple SELECT

# Data Manipulation Language(DML)

- Are used for managing data:
  - SELECT
    - retrieve data from the a database
  - INSERT
    - insert data into a table
  - UPDATE
    - updates existing data within a table
  - DELETE
    - deletes all records from a table

# SQL query

- An SQL query is an SQL statement, which specifies a subset of the data in the database
- A subset in terms of
  - Tables
  - Fields
  - Conditions on fields

# HotelDB

- HOTEL: (Hotel No, Name, Address)
- ROOM: (Room No, Hotel No, Types, Price)
- BOOKING: (BokingID, Hotel\_No, Guest\_No, Date\_From, Date\_To, Room\_No)
- GUEST: (Guest No, Name, Address)

# Table: Guest

<b>Guest_No</b>	<b>Name</b>	<b>Address</b>
1	Eva	Paradisvej 3, 1111 Bispeborg
2	Adam	Paradisvej 7, 1111 Bispeborg
3	Goeg	Sunset Blvd. 8, 2222 Hjemby
4	Gokke	Sunset Blvd. 8, 2222 Hjemby
5	Fy	Klovnevej 87, 3333 Lilleby
6	Bi	Bredgade 198, 3333 Lilleby
7	Romeo	Kaerlighedstunellen 1, 4444 Borgerslev
8	Julie	Kaerlighedstunellen 2, 4444 Borgerslev
9	Godzilla	Dommervænget 16A, 4000 Roskilde
10	KingKong	Hyrdevænget 38, 4000 Roskilde

# SQL query

- The most basic SQL query looks like:

```
SELECT <fieldlist>  
FROM <tablename>
```

Which **fields**  
do I want?  
\* Means all  
fields

From what **table**  
do I want the fields

# SELECT Guest\_no, Name FROM Guest

Guest_No	Name	Address
1	Eva	Paradisvej 3, 1111 Bispeborg
2	Adam	Paradisvej 7, 1111 Bispeborg
3	Goeg	Sunset Blvd. 8, 2222 Hjemby
4	Gokke	Sunset Blvd. 8, 2222 Hjemby
5	Fy	Klovnevej 87, 3333 Lilleby
6	Bi	Bredgade 198, 3333 Lilleby
7	Romeo	Kaerlighedstunellen 1, 4444 Borgerslev
8	Julie	Kaerlighedstunellen 2, 4444 Borgerslev
9	Godzilla	Dommervænget 16A, 4000 Roskilde
10	KingKong	Hyrdevænget 38, 4000 Roskilde

# SELECT \* FROM Guest

Guest_No	Name	Address
1	Eva	Paradisvej 3, 1111 Bispeborg
2	Adam	Paradisvej 7, 1111 Bispeborg
3	Goeg	Sunset Blvd. 8, 2222 Hjemby
4	Gokke	Sunset Blvd. 8, 2222 Hjemby
5	Fy	Klovnevej 87, 3333 Lilleby
6	Bi	Bredgade 198, 3333 Lilleby
7	Romeo	Kaerlighedstunellen 1, 4444 Borgerslev
8	Julie	Kaerlighedstunellen 2, 4444 Borgerslev
9	Godzilla	Dommervænget 16A, 4000 Roskilde
10	KingKong	Hyrdevænget 38, 4000 Roskilde



# Eliminating duplicate rows - DISTINCT

- The keyword DISTINCT filters / removes duplicate rows from the result

# SQL query - ordering

- In a query producing a large result, it may be beneficial to **order** the result
- SQL allows us to order the result by any of the fields in the result
- We use the keyword **ORDER BY**

# SQL query - ordering

**SELECT** <fieldlist>  
**FROM** <tablename>  
**WHERE** <condition>  
**ORDER BY** <fieldname>

Which **fields**  
do I want

From what **table**  
do I want the fields

What **conditions**  
must the fields fulfill

What **order** are the  
results sorted in

# HotelDB

HOTEL: (Hotel\_No, Name, Address)

ROOM: (Room\_No, Hotel\_No, Types,  
Price)

BOOKING: (Hotel\_No, Guest\_No,  
Date\_From, Date\_To, Room\_No)

GUEST: (Guest\_No, Name, Address)

# SQL query - ordering

```
SELECT * FROM Hotel  
ORDER BY Name;
```

```
SELECT * FROM Guest ORDER BY Address;
```

# SQL query - ordering

- We can even specify more than one field for ordering – secondary fields used if primary fields are identical
- We can choose between descending and ascending order, using the keywords **DESC** and **ASC**, respectively

```
SELECT Guest_No, Date_From, Room_no  
FROM Booking  
ORDER BY Guest_No DESC, Date_From ASC;
```

# Exercises – SQL simple SELECT

- Formulate queries in order to retrieve the below data:
  - Get all hotels
  - Show names of all hotels
  - Get all bookings
  - Show names of all guests