



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG



CHAI

Humanities-Centered AI

Databases

**Malte Luttermann – Institute for Humanities-Centered
Artificial Intelligence (CHAI)**

November 20, 2025

Topics for Today

- Structured Query Language (SQL)

Acknowledgement

- Some of the upcoming content is taken from the following lecture (and corresponding exercises) and has been translated and partially modified:
 - PD Dr. habil. Özgür L. Özçep: »Datenbanken«

SQL – Comprehension Questions

Are the following statements correct?

- »If a new row is inserted into a table, columns may remain undefined.«
- »Views in SQL are shortcuts for queries.«
- »In an SQL query, the attributes in the output can be renamed.«

SQL – Comprehension Questions

Are the following statements correct?

- »If a new row is inserted into a table, columns may remain undefined.«
 - Correct. If the schema definition does not enforce that the attribute is not NULL, undefined values are replaced by predefined default values or NULL.
- »Views in SQL are shortcuts for queries.«

- »In an SQL query, the attributes in the output can be renamed.«

SQL – Comprehension Questions

Are the following statements correct?

- »If a new row is inserted into a table, columns may remain undefined.«
 - Correct. If the schema definition does not enforce that the attribute is not NULL, undefined values are replaced by predefined default values or NULL.
- »Views in SQL are shortcuts for queries.«
 - Correct. Views enable you to assign names to query modules, allowing you to access them more quickly.
- »In an SQL query, the attributes in the output can be renamed.«

SQL – Comprehension Questions

Are the following statements correct?

- »If a new row is inserted into a table, columns may remain undefined.«
 - Correct. If the schema definition does not enforce that the attribute is not NULL, undefined values are replaced by predefined default values or NULL.
- »Views in SQL are shortcuts for queries.«
 - Correct. Views enable you to assign names to query modules, allowing you to access them more quickly.
- »In an SQL query, the attributes in the output can be renamed.«
 - Correct. The `as` keyword allows renaming attributes in the output.

SQL Queries

Let the following relational schemas be given:

- *Supplier* = { SupplierId, *Company*, *ContactPerson*, *Street*, *City*, *Telephone* }
- *Category* = { CategoryId, *CategoryName*, *Description* }
- *Order* = { OrderId, *OrderDate*, *CustomerId* }
- *OrderDetails* = { OrderId, ItemId, *Quantity* }
- *Customer* = { CustomerId, *Company*, *ContactPerson*, *Street*, *City*, *Telephone* }
- *Item* = { ItemId, *ItemName*, *SupplierId*, *CategoryId*, *DeliveryUnit* }
- *ItemPrice* = { ItemId, *Price* }

Formulate a query in SQL to determine which suppliers (wanted: *Company*, *ContactPerson*) have their address on Warburgstrasse in Hamburg.

SQL Queries

Formulate a query in SQL to determine which suppliers (wanted: *Company*, *ContactPerson*) have their address on Warburgstrasse in Hamburg.

```
1 SELECT Company, ContactPerson
2 FROM Supplier
3 WHERE Street = 'Warburgstrasse' AND City = 'Hamburg';
```

SQL Queries

Let the following relational schemas be given:

- *Supplier* = { SupplierId, Company, ContactPerson, Street, City, Telephone }
- *Category* = { CategoryId, CategoryName, Description }
- *Order* = { OrderId, OrderDate, CustomerId }
- *OrderDetails* = { OrderId, ItemId, Quantity }
- *Customer* = { CustomerId, Company, ContactPerson, Street, City, Telephone }
- *Item* = { ItemId, ItemName, SupplierId, CategoryId, DeliveryUnit }
- *ItemPrice* = { ItemId, Price }

Formulate a query in SQL to determine how many different customers have already ordered the item »gingerbread«.

SQL Queries

Formulate a query in SQL to determine how many different customers have already ordered the item »gingerbread«.

```
1 SELECT COUNT(DISTINCT CustomerId)
2 FROM Order O, Item I, OrderDetails D
3 WHERE D.ItemId = I.ItemId
4       AND O.OrderId = D.OrderId
5       AND ItemName = 'gingerbread';
```

SQL Queries

Let the following relational schemas be given:

- *Supplier* = { SupplierId, Company, ContactPerson, Street, City, Telephone }
- *Category* = { CategoryId, CategoryName, Description }
- *Order* = { OrderId, OrderDate, CustomerId }
- *OrderDetails* = { OrderId, ItemId, Quantity }
- *Customer* = { CustomerId, Company, ContactPerson, Street, City, Telephone }
- *Item* = { ItemId, ItemName, SupplierId, CategoryId, DeliveryUnit }
- *ItemPrice* = { ItemId, Price }

Formulate a query in SQL to determine the name of the supplier who delivered the items that sold in the highest quantities.

SQL Queries

Formulate a query in SQL to determine the name of the supplier who delivered the items that sold in the highest quantities.

```
1 SELECT DISTINCT Company
2 FROM Supplier S, Item I, OrderDetails D,
3     (SELECT MAX(Quantity) AS MaxNum FROM OrderDetails) AS MaxTmp
4 WHERE I.SupplierId = S.SupplierId
5        AND I.ItemId = D.ItemId
6        AND D.Quantity = MaxNum;
```

SQL Queries

Let the following relational schemas be given:

- *Supplier* = { SupplierId, Company, ContactPerson, Street, City, Telephone }
- *Category* = { CategoryId, CategoryName, Description }
- *Order* = { OrderId, OrderDate, CustomerId }
- *OrderDetails* = { OrderId, ItemId, Quantity }
- *Customer* = { CustomerId, Company, ContactPerson, Street, City, Telephone }
- *Item* = { ItemId, ItemName, SupplierId, CategoryId, DeliveryUnit }
- *ItemPrice* = { ItemId, Price }

Formulate a query in SQL to increase the price of all items by 5%.

SQL Queries

Formulate a query in SQL to increase the price of all items by 5%.

```
1 UPDATE ItemPrice
2 SET Price = Price * 1.05
```

SQL Queries

Let the following relational schemas be given:

- *Supplier* = { SupplierId, Company, ContactPerson, Street, City, Telephone }
- *Category* = { CategoryId, CategoryName, Description }
- *Order* = { OrderId, OrderDate, CustomerId }
- *OrderDetails* = { OrderId, ItemId, Quantity }
- *Customer* = { CustomerId, Company, ContactPerson, Street, City, Telephone }
- *Item* = { ItemId, ItemName, SupplierId, CategoryId, DeliveryUnit }
- *ItemPrice* = { ItemId, Price }

Formulate a query in SQL to insert a new customer »Best Supplies GmbH« located on Main street 5 in Berlin with ID 123, contact person Anna Schmidt, and phone number 123456 into the database.

SQL Queries

Formulate a query in SQL to insert a new customer »Best Supplies GmbH« located on Main street 5 in Berlin with ID 123, contact person Anna Schmidt, and phone number 123456 into the database.

```
1 INSERT INTO Customer
2 VALUES (123, 'Best Supplies GmbH', 'Anna Schmidt', 'Main street
    5', 'Berlin', '123456');
```

SQL Queries

Let the following relational schemas be given:

- *Supplier* = { SupplierId, Company, ContactPerson, Street, City, Telephone }
- *Category* = { CategoryId, CategoryName, Description }
- *Order* = { OrderId, OrderDate, CustomerId }
- *OrderDetails* = { OrderId, ItemId, Quantity }
- *Customer* = { CustomerId, Company, ContactPerson, Street, City, Telephone }
- *Item* = { ItemId, ItemName, SupplierId, CategoryId, DeliveryUnit }
- *ItemPrice* = { ItemId, Price }

The order 42 has been canceled. Formulate a query in SQL to delete all information related to this order from the database.

SQL Queries

The order 42 has been canceled. Formulate a query in SQL to delete all information related to this order from the database.

```
1 DELETE FROM OrderDetails
2 WHERE OrderId = 42;
3
4 DELETE FROM Order
5 WHERE OrderId = 42;
```