Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CHAI
Humanities-Centered AI

Databases

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

December 4, 2025

Version from December 3, 2025

Topics for Today

- Structured Query Language (SQL)
- Architecture of Database Management Systems (DBMS)

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?

- »The use of `CHECK` enables assertions to be made about a table.«
- »The result of an SQL query might contain duplicate rows.«
- »The `HAVING` clause is used to filter rows before grouping them.«
- »Assertions are not required, as all functionalities also work using checks.«

SQL – Comprehension Questions

Are the following statements correct?

- »The use of `CHECK` enables assertions to be made about a table.«
  - Correct. For example, `CHECK (age >= 18)` requires that every new entry that is entered must have an age of at least 18.
- »The result of an SQL query might contain duplicate rows.«
- »The `HAVING` clause is used to filter rows before grouping them.«
- »Assertions are not required, as all functionalities also work using checks.«

SQL – Comprehension Questions

Are the following statements correct?

- »The use of `CHECK` enables assertions to be made about a table.«
  - Correct. For example, `CHECK (age >= 18)` requires that every new entry that is entered must have an age of at least 18.
- »The result of an SQL query might contain duplicate rows.«
  - Correct. To eliminate duplicates, one can use the `DISTINCT` keyword.
- »The `HAVING` clause is used to filter rows before grouping them.«
- »Assertions are not required, as all functionalities also work using checks.«

SQL – Comprehension Questions

Are the following statements correct?

- »The use of `CHECK` enables assertions to be made about a table.«
  - Correct. For example, `CHECK (age >= 18)` requires that every new entry that is entered must have an age of at least 18.
- »The result of an SQL query might contain duplicate rows.«
  - Correct. To eliminate duplicates, one can use the `DISTINCT` keyword.
- »The `HAVING` clause is used to filter rows before grouping them.«
  - Incorrect. The `HAVING` clause is filters groups after the `GROUP BY` operation.
- »Assertions are not required, as all functionalities also work using checks.«

SQL – Comprehension Questions

Are the following statements correct?

- »The use of `CHECK` enables assertions to be made about a table.«
  - Correct. For example, `CHECK (age >= 18)` requires that every new entry that is entered must have an age of at least 18.
- »The result of an SQL query might contain duplicate rows.«
  - Correct. To eliminate duplicates, one can use the `DISTINCT` keyword.
- »The `HAVING` clause is used to filter rows before grouping them.«
  - Incorrect. The `HAVING` clause is filters groups after the `GROUP BY` operation.
- »Assertions are not required, as all functionalities also work using checks.«
  - Incorrect. Assertions are required to create cross-table assurances. Constraints only refer to one table.

SQL – Constraints

Let the following relational schemas be given:

- $Customer = \{\underline{CustomerId}, FirstName, LastName\}$
- $Drink = \{\underline{DrinkId}, DrinkName, Price, Type\}$
- $Order = \{\underline{OrderId}, CustomerId\}$
- $OrderDetails = \{\underline{OrderId}, \underline{DrinkId}, Quantity\}$

Add a constraint that ensures that each drink has a type of either »Beer«, »Cocktail«, or »SoftDrink«.

SQL – Constraints

Add a constraint that ensures that each drink has a type of either »Beer«, »Cocktail«, or »SoftDrink«.

```
1 ALTER TABLE Drink
2 ADD CONSTRAINT EnsureType
3 CHECK (Type IN VALUES('Beer', 'Cocktail', 'SoftDrink'));
```

SQL – Constraints

Let the following relational schemas be given:

- $Customer = \{\underline{CustomerId}, FirstName, LastName\}$
- $Drink = \{\underline{DrinkId}, DrinkName, Price, Type\}$
- $Order = \{\underline{OrderId}, CustomerId\}$
- $OrderDetails = \{\underline{OrderId}, \underline{DrinkId}, Quantity\}$

Add a constraint that ensures that there is at least one drink for each type (»Beer«, »Cocktail«, »SoftDrink«).

SQL – Constraints

Add a constraint that ensures that there is at least one drink for each type (»Beer«, »Cocktail«, »SoftDrink«).

```
ALTER TABLE Drink
ADD CONSTRAINT EnsureAtLeastOneDrinkPerType
CHECK(
  EXISTS(SELECT * FROM Drink WHERE Type = 'Beer') AND
  EXISTS(SELECT * FROM Drink WHERE Type = 'Cocktail') AND
  EXISTS(SELECT * FROM Drink WHERE Type = 'SoftDrink')
);
```

Architecture – Comprehension Questions

Are the following statements correct?

- »If a page is no longer needed in the buffer, it is always written back to the hard disk.«
- »The page size is always the maximum table size in a database.«
- »The record identifier (RID) is another name for the primary key of a table.«

Architecture – Comprehension Questions

Are the following statements correct?

- »If a page is no longer needed in the buffer, it is always written back to the hard disk.«
  - Incorrect. Only if the page has been modified. If this is the case, the dirty bit is set.
- »The page size is always the maximum table size in a database.«
- »The record identifier (RID) is another name for the primary key of a table.«

Architecture – Comprehension Questions

Are the following statements correct?

- »If a page is no longer needed in the buffer, it is always written back to the hard disk.«
  - Incorrect. Only if the page has been modified. If this is the case, the dirty bit is set.
- »The page size is always the maximum table size in a database.«
  - Incorrect. Tables can be significantly larger than a predetermined page size.
- »The record identifier (RID) is another name for the primary key of a table.«

Architecture – Comprehension Questions

Are the following statements correct?

- »If a page is no longer needed in the buffer, it is always written back to the hard disk.«
  - Incorrect. Only if the page has been modified. If this is the case, the dirty bit is set.
- »The page size is always the maximum table size in a database.«
  - Incorrect. Tables can be significantly larger than a predetermined page size.
- »The record identifier (RID) is another name for the primary key of a table.«
  - Incorrect. The RID uniquely identifies rows in a database at the physical level. The primary key identifies (as a combination of attributes) each row in a table in an abstract sense.

Architecture – Memory Hierarchy

Explain why a memory hierarchy is necessary, i.e., why there are several types of memory organised in the form of a pyramid.

	Capacity	Latency
CPU (with registers)	Bytes	< 1 ns
Cache memory	Kilo-/Mega-Bytes	< 10 ns
Main memory	Giga-Bytes	20-100 ns
Flash memory	Giga/Tera-Bytes	30-250 ms
Hard disk	Tera-Bytes	3-10 ms
Tape machine	Peta-Bytes	varying

Architecture – Memory Hierarchy

Explain why a memory hierarchy is necessary, i.e., why there are several types of memory organised in the form of a pyramid.

- A single memory cannot simultaneously meet all requirements:
  - Access time,
  - data transfer rate,
  - price,
  - size, and
  - energy consumption.
- The memory hierarchy creates the illusion of a kind of ideal memory for the processor that is fast, inexpensive, small and energy-efficient

Architecture – Sequential vs. Random Access

Explain the difference between sequential and random access.

Architecture – Sequential vs. Random Access

Explain the difference between sequential and random access.

- Random access:
  - The desired track is searched for again for each block (search time), the correct data block is waited for (rotation delay) and then the data is read (transfer time).
  - The blocks are searched for independently of each other.
- Sequential access:
  - We look at n contiguous (sequential) blocks.
  - The first block is searched for, then we read the next $n - 1$ blocks (transfer time) and if the blocks are spread across multiple tracks, track changes occur.

Architecture – Sequential vs. Random Access

What factors determine access times for sequential and random access? Please provide the specific formula for each case.

Architecture – Sequential vs. Random Access

- Random access:

$$t_{random\ access} = n(t_s + t_r + t_{tr})$$

  - n: Number of blocks to be read
  - t_s: Movement of arms to the desired track (seek time)
  - t_r: Waiting time for the desired block to rotate under the arm (rotation time)
  - t_{tr}: Read time or write time (transfer time)

- Sequential access:

$$T_{sequential\ access} = t_s + t_r + n \cdot t_{tr} + m \cdot t_{s,track-to-track}$$

  - m: Number of track changes (number of tracks occupied by file to be read)
  - $t_{s,track-to-track}$: Time to move from one track to the next

Architecture – Replacement Strategies

When should the random replacement strategy be used? Give an example and explain in more detail why the other strategies are not appropriate in this case.

Architecture – Replacement Strategies

When should the random replacement strategy be used? Give an example and explain in more detail why the other strategies are not appropriate in this case.

- Access patterns for the cache are unknown
- The cache should displace pages independently of access patterns
- Any other strategy would create a bias, which must be avoided