

# DB/DWH DBMS

DBMS → E.F. Codd → Turing → 0. Niklaus Wirth (Pascal)

SQL → DDL - Definition, DML - Manipulation, DCL - Control, DQL - Query

RA → Relational → Operatoren, Funktionen → Operatoren, Anfragen → Rel → another Rel

Integrität → Semantische, Operatoren, Referentielle → FK, PK, Aggregates

Aggregieren, Gruppieren, Umbenennung, Sortierung, Rangfolge → D, J, P, A, M, S

DDL - CREATE, ALTER, DROP, DML - INSERT, UPDATE, DELETE, DQL - SELECT, FROM, WHERE, DCL - GRANT, REVOKE

SQL → Bags (multimengen), EBNF → extended Backus-Naur Form, DROP DATABASE IF EXISTS, CREATE DATABASE, USE, CREATE TABLE, INSERT INTO, VALUES, FROM, WHERE, GROUP BY, HAVING, SELECT, ORDER BY

NULL 3N (3-value-Logic) → True, False, Unknown

Schema → Kopf, unversch, Tupeln → Zeilen, versch, Attribut → Bez. 1 Domäne (Datentyp), Semantische, Operatoren, Referentielle → FK, PK, Aggregates

Relational → Operatoren, Funktionen → Operatoren, Anfragen → Rel → another Rel

Integrität → Semantische, Operatoren, Referentielle → FK, PK, Aggregates

Aggregieren, Gruppieren, Umbenennung, Sortierung, Rangfolge → D, J, P, A, M, S

DDL - CREATE, ALTER, DROP, DML - INSERT, UPDATE, DELETE, DQL - SELECT, FROM, WHERE, DCL - GRANT, REVOKE

SQL → Bags (multimengen), EBNF → extended Backus-Naur Form, DROP DATABASE IF EXISTS, CREATE DATABASE, USE, CREATE TABLE, INSERT INTO, VALUES, FROM, WHERE, GROUP BY, HAVING, SELECT, ORDER BY

NULL 3N (3-value-Logic) → True, False, Unknown

Schema → Kopf, unversch, Tupeln → Zeilen, versch, Attribut → Bez. 1 Domäne (Datentyp), Semantische, Operatoren, Referentielle → FK, PK, Aggregates

Relational → Operatoren, Funktionen → Operatoren, Anfragen → Rel → another Rel

Integrität → Semantische, Operatoren, Referentielle → FK, PK, Aggregates

Aggregieren, Gruppieren, Umbenennung, Sortierung, Rangfolge → D, J, P, A, M, S

DDL - CREATE, ALTER, DROP, DML - INSERT, UPDATE, DELETE, DQL - SELECT, FROM, WHERE, DCL - GRANT, REVOKE

SQL → Bags (multimengen), EBNF → extended Backus-Naur Form, DROP DATABASE IF EXISTS, CREATE DATABASE, USE, CREATE TABLE, INSERT INTO, VALUES, FROM, WHERE, GROUP BY, HAVING, SELECT, ORDER BY

NULL 3N (3-value-Logic) → True, False, Unknown

Schema → Kopf, unversch, Tupeln → Zeilen, versch, Attribut → Bez. 1 Domäne (Datentyp), Semantische, Operatoren, Referentielle → FK, PK, Aggregates

Relational → Operatoren, Funktionen → Operatoren, Anfragen → Rel → another Rel

Integrität → Semantische, Operatoren, Referentielle → FK, PK, Aggregates

Aggregieren, Gruppieren, Umbenennung, Sortierung, Rangfolge → D, J, P, A, M, S

DDL - CREATE, ALTER, DROP, DML - INSERT, UPDATE, DELETE, DQL - SELECT, FROM, WHERE, DCL - GRANT, REVOKE

SQL → Bags (multimengen), EBNF → extended Backus-Naur Form, DROP DATABASE IF EXISTS, CREATE DATABASE, USE, CREATE TABLE, INSERT INTO, VALUES, FROM, WHERE, GROUP BY, HAVING, SELECT, ORDER BY

NULL 3N (3-value-Logic) → True, False, Unknown

Schema → Kopf, unversch, Tupeln → Zeilen, versch, Attribut → Bez. 1 Domäne (Datentyp), Semantische, Operatoren, Referentielle → FK, PK, Aggregates

Relational → Operatoren, Funktionen → Operatoren, Anfragen → Rel → another Rel

Integrität → Semantische, Operatoren, Referentielle → FK, PK, Aggregates

# DWH → Integration unterschiedlicher Daten in ein System (Golden Source)

OLAP → Online Analytical Processing → Modellfokus, Subject-oriented, Integrated, Time-variant, Non-volatile, OLTP → Online Transaction Processing, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed

Star Schema → Data → Staging → Data Mart, OLAP → big queries w/ ++ joins, Updates, redundanz, de-normalisierung, i.e. verteilten speed