Terminologies and Definitions

Database

An organized collection of data, typically stored and accessed electronically.

Database Management System (DBMS)

Software that stores, manages, and provides controlled, concurrent, and efficient access to databases; supports recovery, security, and data integrity.

Relational Model

A data model where data is represented as relations (tables), each a set of tuples (rows) over a fixed set of attributes (columns), queried with a declarative language (e.g., SQL).

Relation

A set of tuples defined over a schema (attribute names and their domains). Duplicate tuples are not allowed in the pure relational model.

Tuple

A row in a relation; an ordered mapping from attribute names to values in their domains.

Attribute

A named column of a relation, associated with a domain (type) of allowed values.

Domain

The set of permissible values for an attribute (e.g., INTEGER, CHAR, DATE).

Schema

The structural description of data. A relation schema lists a relation name and its attributes (with types). A database schema is a set of relation schemas plus constraints.

Key

A set of attributes that uniquely identifies each tuple in a relation.

Superkey

Any set of attributes that contains a key; i.e., it functionally determines all attributes in the relation.

Candidate Key

A minimal superkey (no proper subset is a superkey).

Primary Key

One chosen candidate key used to identify tuples in a relation; values are unique and (typically) NOT NULL.

Foreign Key

Attribute(s) in one relation that reference the primary key (or a candidate key) of another relation; used to link related tuples.

Referential Integrity

A constraint requiring each foreign key value to match some existing referenced key value (or be NULL, if allowed) in the referenced relation.

NULL

A special marker denoting missing or unknown information in SQL. Comparisons with NULL use three-valued logic (TRUE, FALSE, UNKNOWN).

Entity

A distinguishable object or thing in the modeled domain.

Entity Set

A collection of similar entities (analogous to a relation of tuples).

Relationship

An association among one or more entities.

Relationship Set

A set of relationships of the same type among the same participating entity sets.

Weak Entity

An entity set whose key comes not completely from its own attributes; it depends on a strong entity set via an identifying relationship (often many-one/one-one). Drawn with double rectangle in E/R diagrams.

ISA (is-a) Hierarchy

Specialization/generalization between entity sets (subclass and superclass). Subclass entities inherit attributes and relationships of the superclass.

Aggregation

An E/R modeling construct that treats a relationship (possibly with participating entities) as an abstract higher-level entity so it can participate in other relationships.

Cardinality Constraint

The multiplicity of a relationship: one-to-one (1:1), one-to-many (1:N), or many-to-many (M:N).

Participation Constraint

Whether participation of an entity set in a relationship set is total (every entity participates) or partial (some may not participate).

Relational Algebra

A formal language for querying relations using operators such as selection (σ), projection (π), union ($\cup$), difference ($-$), Cartesian product ($\times$), and renaming ($\rho_{\text{NewTable(NewCol1, NewCol2)}}$); derived operators include join ($\bowtie$), natural join, and intersection ($\cap$).

Selection

A relational algebra operator that filters rows according to a predicate (horizontal subset). In SQL: `WHERE`.

Projection

A relational algebra operator that selects a subset of columns (vertical subset), removing duplicates in the set semantics model.

Join

Combines tuples from two relations based on a join condition. The natural join matches tuples on common attributes with equal values.

Outer Join

Variants of join (LEFT/RIGHT/FULL) that additionally include dangling rows from one or both inputs, padding missing attributes with NULLs.

Grouping and Aggregation

Partitioning result tuples into groups (GROUP BY) and computing aggregate values per group (e.g., COUNT, SUM, AVG, MIN, MAX). HAVING filters groups by aggregate predicates.

Set vs. Bag Semantics

SQL by default uses bag (multiset) semantics for `SELECT-FROM-WHERE`; `DISTINCT` enforces set semantics. Set operations `UNION`/`INTERSECT`/`EXCEPT` are set-based; bag variants use `ALL`.

View

A virtual table defined by a query; its contents are not stored (unless materialized) and are computed when referenced.

Index

An auxiliary data structure that accelerates data access by keys or search predicates. Common choices include B$^+$-tree indexes (ordered, range-friendly) and hash indexes (exact-match efficient).

Constraint

A condition declared in the schema and enforced by the DBMS. Common constraints: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY (referential integrity), CHECK, and assertions.

Data Definition Language (DDL)

SQL subset for defining and modifying schema objects (`CREATE`, `ALTER`, `DROP`).

Data Manipulation Language (DML)

SQL subset for querying and updating data (`SELECT`, `INSERT`, `UPDATE`, `DELETE`).

Functional Dependency (FD)

For relation R, an FD $X \to Y$ (where X and Y are sets of attributes) means that any two tuples that agree on X must also agree on Y. Captures constraints among attributes and is central to schema design.

Attribute Closure

Given a set of FDs $\mathcal{F}$ and attribute set Z, the closure Z^+ is the set of all attributes functionally determined by Z under $\mathcal{F}$. Used to test keys and FD implication.

Armstrong's Axioms

Sound and complete inference rules for FDs: Reflexivity (if $Y \subseteq X$ then $X \to Y$), Augmentation (if $X \to Y$ then $XZ \to YZ$), and Transitivity (if $X \to Y$ and $Y \to Z$ then $X \to Z$).

Key (via FDs)

A set of attributes K is a key of R w.r.t. $\mathcal{F}$ if K^+ contains all attributes of R and K is minimal with this property. A superkey need not be minimal.

Lossless Join Decomposition

A decomposition of relation R into S and T is lossless (w.r.t. constraints such as FDs) if $S \bowtie T$ always yields exactly R (no spurious tuples) for any legal instance of R.

Dependency Preservation

A decomposition preserves dependencies if the union of the projected

FDs on the components implies all original FDs (i.e., one can enforce all FDs without recomputing joins).

Boyce–Codd Normal Form (BCNF)
A relation schema is in BCNF if, for every non-trivial FD $X \to Y$ that holds, X is a superkey. Eliminates all redundancy due to FDs.

Third Normal Form (3NF)
A relation schema is in 3NF if, for every FD $X \to Y$, either (i) the FD is trivial ($Y \subseteq X$), or (ii) X is a superkey, or (iii) every attribute in Y is prime (appears in some candidate key).

Second Normal Form (2NF)
Every non-prime attribute is fully functionally dependent on the whole of every candidate key (i.e., no partial dependency on a proper subset of a candidate key).

First Normal Form (1NF)
All attribute values are atomic (no repeating groups, arrays, or nested relations in a single column).

Multivalued Dependency (MVD)
A constraint $X \twoheadrightarrow Y$ stating that, for a fixed X value, the Y values are independent of the rest of the attributes; captures independent multi-valued facts.

Fourth Normal Form (4NF)
A relation schema is in 4NF if, for every non-trivial MVD $X \twoheadrightarrow Y$, X is a superkey. Stronger than BCNF.

SQL
Structured Query Language; the standard declarative language for defining, querying, and manipulating relational data.

`SELECT–FROM–WHERE`
Core SQL query template: `FROM` lists input tables, `WHERE` filters rows, `SELECT` chooses output columns and expressions; `DISTINCT` removes duplicates; `ORDER BY` specifies ordering.

`JOIN` (SQL)
`INNER JOIN` returns matching rows according to an `ON` (or `USING`) condition; `NATURAL JOIN` equates common-name attributes; `OUTER JOIN`s additionally include dangling rows, padded with `NULL`s.

`GROUP BY` (SQL)
Forms groups of rows sharing the same values on grouping columns; aggregates are computed per group; only grouping columns and aggregates may appear in `SELECT` (SQL standard). `HAVING` filters groups by aggregate predicates.

Constraints in SQL
`NOT NULL` (attribute cannot be `NULL`); `UNIQUE` (all non-`NULL` values distinct); `PRIMARY KEY` (`UNIQUE` + `NOT NULL`, at most one per table); `FOREIGN KEY` (referential integrity between tables); `CHECK` (row-level predicate); `ASSERTION` (database-level predicate).

Referential Actions
Actions on foreign key constraint violation or updates: RESTRICT/NO ACTION (reject), CASCADE (propagate), SET NULL (replace with NULL), SET DEFAULT (replace with default), depending on DBMS support.

Additional Concepts from Lectures 1–8

Data Independence
The separation between application programs and data organization. Logical data independence means schema changes (e.g., new attributes/tables) need not break applications. Physical data independence means changes to file structures or indexes don't affect logical schemas or applications.

NoSQL
A family of non-relational database systems emphasizing scalability, availability, and flexible schemas. Categories include key–value stores, document stores, wide-column stores, and graph databases; often trade strong consistency for eventual consistency.

Key–Value Store
A NoSQL model storing arbitrary values addressed by keys (e.g., Redis, Riak). APIs commonly include get/put operations; schema is application-defined.

Document Store
Stores semi-structured documents (e.g., JSON), enabling nested fields and flexible schemas (e.g., MongoDB, Couchbase). Supports ad-hoc queries on document fields.

Wide-Column Store
Stores sparse, column-family oriented data across distributed nodes (e.g., Bigtable, HBase, Cassandra). Optimized for large-scale reads/writes.

Graph Database
Models data as nodes and edges with properties (e.g., Neo4j). Efficient for traversals and graph queries.

Eventual Consistency
A consistency model in which replicas converge to the same state in the absence of further updates. Allows higher availability and partition tolerance at the cost of temporary inconsistencies.

Bag vs. Set Semantics
SQL SELECT produces bags (multisets) by default; duplicates persist unless DISTINCT is used. Set operations UNION/INTERSECT/EXCEPT remove duplicates; ALL variants keep duplicates.

Set Operations (SQL)
UNION combines rows from two queries; INTERSECT returns common rows; EXCEPT (MINUS) returns rows in the first query not in the second. By default, they use set semantics (deduplicate results).

Subquery
A query nested inside another. Uncorrelated subqueries can be evaluated independently; correlated subqueries depend on outer query rows. Commonly used with IN, EXISTS, ANY/ALL, or as derived tables.

ORDER BY
Sorts query results by one or more expressions (ASC/DESC). Not part of relational algebra; affects output presentation, not the relation itself.

Natural Join
A join equating all pairs of same-named attributes from the two input relations and projecting one copy of each such attribute in the result.

`WITH` (Common Table Expressions, CTEs)
Defines temporary named subqueries usable in a following statement. Enables readable decomposition and recursion. Syntax: `WITH name AS (subquery) [, ...]  SELECT ...`. Recursive CTEs use `WITH RECURSIVE` and a base+recursive part combined with `UNION [ALL]`.

OLTP (Online Transaction Processing)
Workloads with many short, concurrent read/write transactions over relatively small portions of data; require strict ACID guarantees, low latency, and high throughput; schemas are typically normalized.

OLAP (Online Analytical Processing)
Read-heavy analytical workloads scanning large portions of data, performing complex aggregations across many rows; emphasize throughput and complex queries over strict per-row transaction guarantees; often use star/snowflake schemas and columnar storage.

NewSQL
Database systems that aim to provide SQL and full ACID transactions with horizontal scalability comparable to NoSQL systems, via techniques like distributed consensus/replication and partitioning.

MapReduce (Map, Shuffle, Reduce)
A batch processing model for large datasets: Map transforms input into key–value pairs; Shuffle groups/sorts by key and routes data; Reduce aggregates or combines values per key. Suited for scalable, fault-tolerant parallel processing.

Parallel DBMS
A shared-nothing, massively parallel relational system that partitions data across nodes and executes queries in parallel (e.g., parallel scans, joins, aggregations) with exchange operators for data redistribution. Provides SQL, cost-based optimization, and parallel execution plans.

Lakehouse
A data architecture that combines the openness and low-cost storage of a data lake with the governance, schema enforcement, and performance of a data warehouse, often adding ACID transactions and indexing on files stored in object storage.

Schema vs. Instance
The schema is the database's structural metadata (tables, attributes, types, and constraints); an instance is the actual current contents (the set of tuples) that conforms to the schema at a point in time.