15. Review 1 (Conceptual Level)
Main Topics

• ER model
• Relational model
• SQL
• Functional Dependencies
• Database design
Symbols of ER Diagram

- **E**: Entity Set
- **A**: Attribute
- **E**: Weak Entity Set
- **A**: Multivalued Attribute
- **R**: Relationship Set
- **A**: Derived Attribute
- **R**: Identifying Relationship Set for Weak Entity Set
- **R**: Total Participation of Entity Set in Relationship
- **E**: Discriminating Attribute of Weak Entity Set
- **A**: Primary Key
Symbols of ER Diagram (Cont.)

- **Many to Many Relationship**
- **Many to One Relationship**
- **One to One Relationship**
- **Role Name**
- **Role Indicator**
- **Cardinality Limits**
- **ISA (Specialization or Generalization)**
- **Total Generalization**
- **Disjoint Generalization**
Relational Model

- Relations: schema and instances
- Attributes: single-valued, domain, keys
- Set of records: no duplicates, no order
- Formal query languages:
  - Relational algebra (RA)
- Automatic conversion between relational and ER
Relational Schema of the Bank

branch (branch-name, branch-city, assets)
customer (customer-name, customer-street, customer-city)
account (account-number, branch-name, balance)
loan (loan-number, branch-name, amount)

depositor (customer-name, account-number)
borrower (customer-name, loan-number)

Keys are underlined and foreign keys are in italics.
Relational Algebra

- Basic operations:
  - Selection (σ)
  - Projection (π)
  - Cross-product (×)
  - Set-difference (-)
  - Union (∪)
  - Renaming (ρ)

- Additional operations:
  - Intersection, join, division
SQL Query Block

SELECT select-clause
FROM from-clause
[WHERE where-clause]
[ORDER BY order-by-expression]
[GROUP BY group-by-attributes]
[HAVING condition-for-each-group]]

Query blocks may be nested in FROM and WHERE; may be connected using UNION, INTERSECT, and EXCEPT.
SQL Features

- Duplicates: DISTINCT.
- Aggregation queries (e.g., max, sum) return a single value, unless there is a group by
- All non-aggregation attributes in SELECT with a GROUP BY must also appear in GROUP BY.
- If a attribute appears in GROUP BY, it may not necessarily appear in SELECT
Functional Dependencies

- The functional dependency $X \rightarrow Y$ holds on $R$ if and only if for any legal relations $r(R)$, whenever any two tuples $t_1$ and $t_2$ of $r$ agree on the attributes $X$, they also agree on the attributes $Y$.
- The set of all functional dependencies logically implied by $F$ is the closure of $F$.
- For computing the closure we use Armstrong’s axioms
  - if $Y \subseteq X$, then $X \rightarrow Y$ (reflexivity)
  - if $X \rightarrow Y$, then $ZX \rightarrow ZY$ (augmentation)
  - if $X \rightarrow Y$, and $Y \rightarrow Z$, then $X \rightarrow Z$ (transitivity)
- Given a set of attributes $X$, the closure of $X$ under $F$ (denoted by $X^+$) is the set of attributes that are functionally determined by $X$ under $F$.
- If $X$ determines all attributes, then it is a superkey. If it is also minimal, then it is a candidate key.
- A canonical cover of $F$ is a “minimal” set of functional dependencies equivalent to $F$, with no redundant dependencies or having redundant attributes.
Normalization

- Normalization is the process of decomposing a relation schema $R$ into fragments (i.e., smaller tables) $R_1, R_2, \ldots, R_n$. The goals are:
  - **Lossless decomposition**: The fragments should contain the same information as the original table. Otherwise decomposition results in information loss.
  - **Dependency preservation**: Dependencies should be preserved within each $R_i$, i.e., otherwise, checking updates for violation of functional dependencies may require computing joins, which is expensive.
  - **Good form**: The fragments $R_i$ should not involve redundancy. A table has redundancy if there is a FD where the LHS is not a key.
Normal Forms

- **BCNF**: for every FD $X \rightarrow Y$, $X$ is a candidate key.
  - There is no redundancy
- **3NF**: a table in BCNF also satisfies 3NF. In addition, 3NF allows FDs where every attribute in $Y$ is prime.
  - There is some redundancy
- **1NF**: every relational table is 1NF because all attribute values are atomic.
BCNF Decomposition Algorithm

Compute $F^+$;
Result = $\{R\}$; $R_0 = R$; $i = 0$;
While (any $X \rightarrow Y$ in $R_i$ violates BCNF) {
    Create a new table $(X, Y)$;
    Result = (Result-$\{R_i\}$) U $\{(R_i - Y)\}$ U $\{(X,Y)\}$;
    $R_{i+1} = (R_i - X)$; $i++$;
}

The final tables are in BCNF. The decomposition is lossless-join, does not have redundancy, but may not preserve dependencies.
Let $F_c$ be a canonical cover for $F$; $i := 0$;

for each $X \rightarrow Y$ in canonical cover $F_c$

    If none of the tables contains $X$ and $Y$
        create $(X,Y)$

    if none of the created tables contains a candidate key
        create a table with any candidate key for $R$;

The final tables are in 3NF. The decomposition is both lossless-join and dependency-preserving, but the tables may have redundancy.