

Functional Dependencies – $X \rightarrow Y$ means X functionally determines Y

- Goal is to define better databases
- To find FD's, can infer from the ones you identify
 - Rules of Inference
 - 1) IR1 (reflexive): if $X \supseteq Y$, then $X \rightarrow Y$, i.e. $\{A, B, C, D\} \rightarrow \{B, C\}$
 - 2) IR2 (augmentation): if $X \rightarrow Y$, then $XZ \rightarrow YZ$
 - 3) IR3 (transitive): if $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$
- Useful for finding candidate keys
 - Given $R(A, B, C, D, E)$ and FD's
 - $\{A, B\} \rightarrow C$
 - $\{A, D\} \rightarrow E$
 - $D \rightarrow E$
 - A, B, D is a **minimal cover** – remove any element and you will no longer have a key.
- **Superkey** – any set of attributes which can uniquely identify a database record
- **Key** – minimal superkey such that the removal of any attribute will cause it to not be a superkey
- **Candidate Keys** – all keys of a relation
- **Normal Forms** – normalizes relational schemas using FD's to eliminate different kinds of anomalies (achieve better database design)
 - **First Normal Form (1NF)**
 - All attributes must be atomic.
 - Eliminates relations within relations; simplifies queries and query languages
 - **Second Normal Form (2NF)**
 - Attribute is **non-prime** if it does not belong to a key; otherwise, it is **prime**
 - A relation is in 2NF if every nonprime attribute A in R is fully functionally dependent on the primary key of R (depends on every attribute/element of key R, can be transitive)
 - **Third Normal Form (3NF)**
 - Every non-prime attribute is fully functionally dependent on every key of R and is nontransitively dependent on every key of R.
 - **Boyce Codd Normal Form (BCNF)**
 - Restriction of 3NF: for all non-trivial FDs $X \rightarrow Y$, X is a key of R (trivial dependency is $X \rightarrow Y$ where $X \supseteq Y$); essentially, everything on left side of a functional dependency must be a key
- **Multivalued Dependencies**
 - In a table R with at least 3 columns, A, B, and C, $R.A \twoheadrightarrow R.B$ (column A multidetermines B) if and only if the set of B-values matching a given pair of A-values and C-values in R depends only on the A-value and is independent of the C-value
 - Always occur in pairs. That is $R.A \twoheadrightarrow R.B$ holds if and only if $R.A \twoheadrightarrow R.C$ also hold.
- **Entity-Relationship Models**
 - Entity: represented by a rectangle
 - Relationship: represented by a diamond; can have attributes, is an association among two or more entities with cardinalities
 - Attributes: represented by a bubble
 - Key Attribute: represented by a bubble with contents underlined
 - Inheritance (subclassing)
- **Turning ER Diagram into Tables**
 - Entity – create a table with the attributes attached
 - Entity – identify the primary key
 - 1:1 relationship – add primary key of one table as a foreign key in the second. Add attributes of

relation.

- 1:M relationship – add primary key from ‘1’ end of relationship as foreign key in ‘M’ end. Add attributes to ‘M’ side.
- M:N relationship – create new table with primary key as the foreign keys from ‘M’ and ‘N’ and attributes as those associated with relationship
- **Relational Algebra**
 - Operands are relations or relation variables
 - Selection, projection, products, joins, theta join, natural join
 - Sets vs. bags – traditionally RA operates on sets, but operations are extended to bags
 - Extended operations ala SQL: Delta (eliminate dups), Tau (sort), Gamma(group), Outerjoin (insert NULLs), extended projection
- **Expression Trees**
 - Useful for query optimization and reasoning about queries using RA
- **SQL**
 - **SELECT,INSERT,UPDATE,DELETE etc.**
 - 3-valued logic: True, False, Unknown
 - True = 1, False = 0, Unknown = $\frac{1}{2}$: X and Y = $\min(x,y)$. X or Y = $\max(x,y)$
- **DB Constraints**
 - Keys, foreign-keys, value-based and tuple-based checks, assertions
- **DB Triggers**
 - Executed upon prespecified conditions