Basic Info

Collection of data elements stored in a computer in a systematic way

Reliable storage and recovery of data

Support for concurrent users

Programmer shouldn't need to know how its stored

Only needs to interact with logical model

Building a database

Start with conceptual model-> Build schemas from this -> Write applications using DBMS -> DBMS takes care of query optimization

SQL

Null rules:

If all are null then its a null result

If only some are null then they are discarded

For count, nulls are counted

Normal Forms

BCNF: For every FD, X->A, A is in X, or X is a superkey for R

Sometimes not dependency preserving

3NF: For every FD, X->A, A is in X, or X is a superkey, or A is a member of some key

1NF: Each entry is a discrete value

3NF decomposition guarantees both lossless and dependency preservation

Transactions

E-R Diagrams

Attributes are only connected to entities and relationships

Keys are attributes that uniquely identify entries

Number of columns in a relationship is the arity

Domain is where the values of an attribute: string, int, etc..

Candidate key: May have several

Primary key: The one key that is chosen to represent a tuple

Super key: A set of fields that include a key

Foreign key: Set of fields in one table that refer to the key in another relation

Entries can only be connected by relationships

Arrow is placed from many to 1 direction

E-R Diagrams (cont)

Weak entities: Can only be identified by the primary key of another entity

Functional Dependencies

Saying one attribute determines another

Armstrong's axioms

Reflexivity, if Y is in X, X->Y

Augmentation, if X->Y, WX->WY

Transitivity, if X->Y and Y->Z, X->Z

Extra's to armstrong's axioms (can be deduced from the axioms)

Union, if X->Y and X->Z, X->YZ

Psuedo-transitivity, if X->Y and WY->Z, XW->Z

Decomposition, if X->Y and Z is in Y, X->Z

Closure of F: Set of all FD's derivable from F

Two FD's are equivalent if their closures are equivalent

Minimal cover, F is minimal if

Every FD in F is of the form X->A where A is a single attribute

For no X->A, is F - {X->A} equivalent to F

For no X->A and subset Z of X is (F - {X->A}) U {X->Z} equivalent to F