Recommended page for Entity Dictionary and Relationship Dictionary

ENTITY DICTIONARY

Entity: Airplane
Description: An aeroplane
Attribute: Flight Number, INT
Attribute: Type VARCHAR(120)
Primary Key: Flgiht Number

Entity: Business Class
Description: The middle range of seating on the flight
Attribute: Credit Card Offer, VARCHAR(120)
Primary Key: Passport, subclass of Traveller

Entity: Economy Class
Description: The low range of seating on the flight
Attribute: Pretzels, Boolean
Primary Key: Passport, subclass of Traveller

Entity: First Class
Description: The high range of seating on the flight
Attribute: Beverage, Boolean
Primary Key: Passport, from superclass Traveller

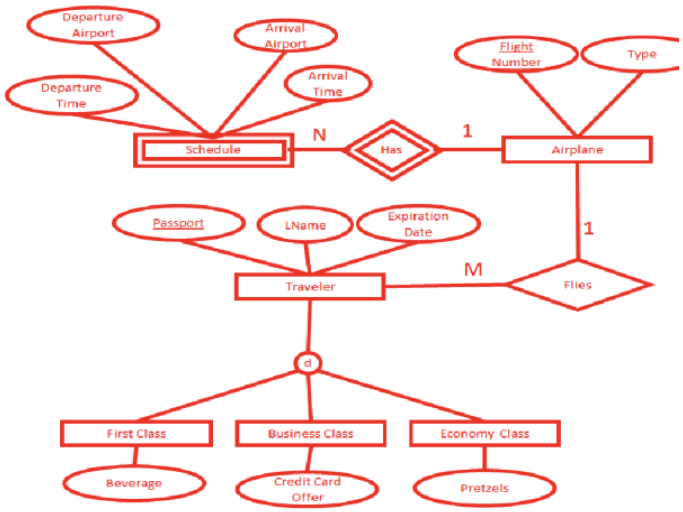
Entity: Schedule
Description: A record of the time and place of the flgiht
Attribute: Arrival Aiprot, VARCHAR(3)
Attribute: Arrival Time, Time
Attribute: Departure Airport, VARCHAR(3)
Attribute: Departure Time, TIME
Primary Key: Weak entity of Flight, 1-to-N cardinality, uses Airplane Flight Number. (Migh

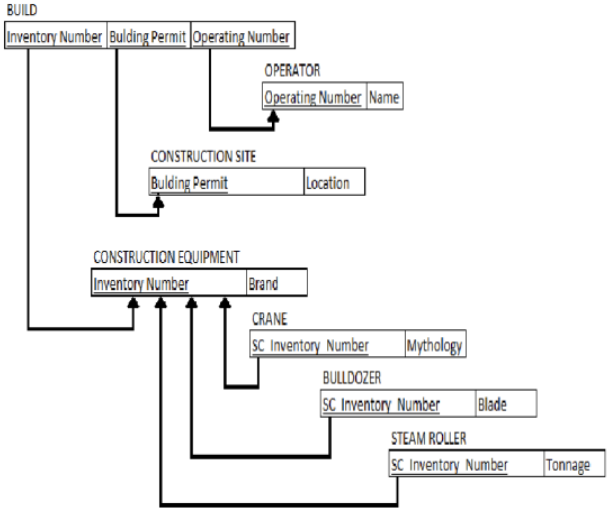
RELATIONSHIP DICTIONARY

Relationship: Flies
Description: A traveller flies on an airplane
Entities: Airplane, Traveller
Participation: Airplane, Partial to Traveller, Partial (Planes can fly without travellers, trav
Cardinality: Airplane, 1 to Traveller, Many

Relationship: Has
Description: An airplane has a schedule for mutliple flights
Entities: Airplane, Schedule
Participation: Airplane, partial to Schedule, Total
Cardinality: Airplane, 1 to Schedule, Many

Recommended page for EER Diagram





Schema Dictionary

BUILD
Primary Key: Inventory_Number, Building_Permit, Operating_Number
Foreign Key: Inventory_Number references Construction Equipment Inventory Number
Foreign Key: Building_Permit references Construction Site.Building Permit
Foreign Key: Operating_Number references Operator.Operating_Number

BULLDOZEZR
Primary Key: SC_Inventory_Number
Foreign Key: SC_Inventory_Number references Construction Equipment.Inventory Number

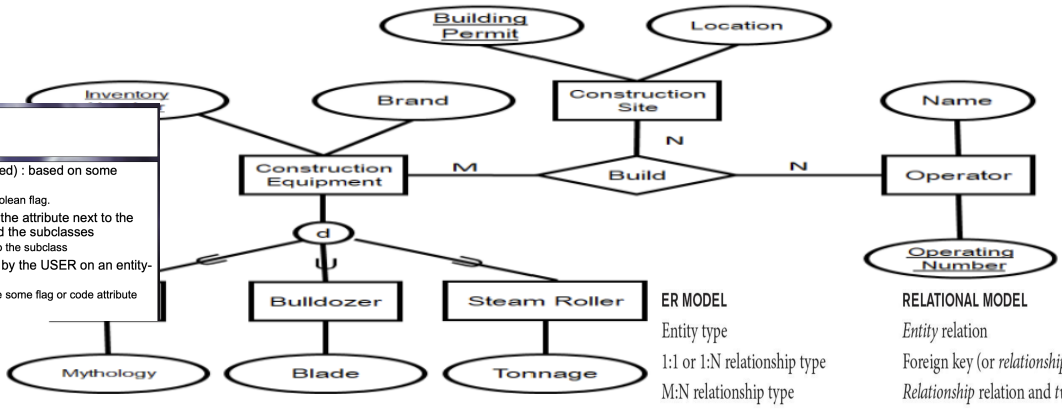
CONSTRUCTION EQUIPMENT
Primary_Key: Inventory_Number

CRANE
Primary Key: SC_Inventory_Number
Foreign Key: Inventory Number references Construction Equipment.Inventory Number

OPERATOR
Primary Key: Operating_Number

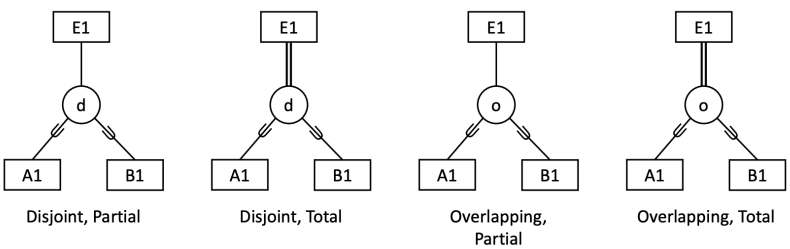
STEAM ROLLER
Primary Key: SC_Inventory_Number
Foreign Key: Inventory Number references Construction Equipment.Inventory Number

Enhanced Entity Relationship Diagram



Specialization: Types

- Predicate-defined (or condition-defined) : based on some predicate.
  - The value stored is in a range or is a Boolean flag.
- Attribute-defined: shows the name of the attribute next to the line drawn from the superclass toward the subclasses
  - The value is an attribute that is unique to the subclass
- User-defined: membership is defined by the USER on an entity-by-entity basis
  - Oh, dear. The design will have to include some flag or code attribute the human being's preference.



ER MODEL	RELATIONAL MODEL
Entity type	*Entity* relation
1:1 or 1:N relationship type	Foreign key (or *relationship* relation)
M:N relationship type	*Relationship* relation and *two* foreign keys
n-ary relationship type	*Relationship* relation and *n* foreign keys
Simple attribute	Attribute
Composite attribute	Set of simple component attributes
Multivalued attribute	Relation and foreign key
Value set	Domain
Key attribute	Primary (or secondary) key

Databases are Boring:**Database**: Standardized structured files for easier data handling.**DBMS**: System with built-in commands to manipulate data. No custom code needed.**NoSQL**: Stands for "Not Only SQL", but relational databases remain foundational.**DBMS Advantages**:Data integrity and security.Data redundancy control.Concurrent access by multiple users.Data recovery and backup.**NoSQL Types**: Document, Key-Value, Columnar, Graph.**Database Architecture**:**Client-Server**: Client = application using data, Server = database storing data.**Data Levels**:(HIGHLVL)**Conceptual**: User's perspective, ER Model.(MID)**Representational**: Abstract tables, Relational Schema Model.(low)**Physical**: Focus on hardware, storage, cost, and speed.**Three-Schema Architecture**: **Logical Independence**: Conceptual schema can change without affecting apps or hardware.**Physical Independence**: Hardware changes don't impact the logical layer or apps.**Database States**:The database is dynamic and always changing.Operations focus on the current "snapshot" or "frame"aka instance.**Data Independence**:Changes at one level don't necessitate changes at other levels.Provides flexibility and robustness against changes.**Normal Forms:Functional Dependency**: Determines which attributes index others; used to identify keys.**1NF**: Single value per cell.**2NF**: All non-key attributes are fully functionally dependent on the primary key.**3NF**: Attributes are functionally dependent on the primary key, not on other attributes.**BCNF**: Every non-trivial functional dependency has a superkey.**4NF**: No multi-valued dependencies on a candidate key.**5NF**: Every join dependency is implied by candidate keys.**Functional Dependencies**:Key in determining organization of data.Important for achieving normalization.**BCNF vs. 3NF**: BCNF is a stronger form of 3NF, where every non-trivial FD has a superkey.**Join Dependency**: How tables can be decomposed and then naturally rejoined without loss.**Relational Algebra**:**Unary Operations**:**SELECT (σ)**: Filters tuples based on a condition.**PROJECT (π)**: Returns specific attributes.**RENAME (ρ)**: Changes attribute names.**Set Operations**:**UNION (∪)**: Combines two sets.**INTERSECTION (∩)**: Returns common elements.**DIFFERENCE (-)**: Removes shared elements.**CARTESIAN PRODUCT (X)**: Creates all possible tuples from two sets.**Difference between FK and FD**: FK (Foreign Key) is a reference from one table to another. FD (Functional Dependency) is an internal reference between two attributes of the same table.**Relational Set Theory**:Operations like UNION and INTERSECTION require "type compatibility" between relations.

a relation looks like a table of values
The data elements in each row represent an instance that correspond to an EER Model's entity or relationship
Each column has a column header that identifies the attribute in that column

ER TO SCHEMA MAPPING

1: Regular entities first, with their attributes

2: Weak entities, with their primary key being a part of a superkey with the owning strong entity's primary key.

3: 1-to-1: If one side of the relationship is a total participation, the primary key of the partial participation is copied as an attribute in the total participation side. If both sides are total (onto and one-to-one), merge the two entities into one entity

4: 1-to-M The primary key of the relationship on the 1-side relational table becomes a foreign key in the N-side of the Relationship. Additional relationship attributes can also be moved to the NSide of relational table.

5:When having an M-to-N relationship, create an independent table. This relational table will have two foreign keys that are the primary keys of the two entities in the relationship. Each entry of the table should be a unique superkey made up of the foreign keys. Any attributes, domains, or constraints of the relationship would exist in this relational table.

6: Multivariate attributes: • The table has the primary key of the owning entity as a foreign key • The primary key of the relational table is the superkey of the foreign key and the attributes.

7: Reference arrows go FROM foreign key TO Primary Key

8: N-ary relationships: Create a new relational table with a foreign key for each entity n the N-ary relationship. The primary key of the relational table is the superkey of the foreign keys. Any additional attributes of the relationship would also be in this relational table.

9: Specializations: 8a)Create a relation table for the superclass that includes all the components (attributes, domains, constraints). • For each subclass, have another table with the remaining attributes with a matching primary key 8b) Create a different relational table for each subclass. • The participation must be total. • And no case of an independent superclass exists. 8c) • The subclass has only one attribute different than the superclass. • Create a relation table for the superclass that includes all the components (attributes, domains, constraints) of all the subclasses. • The attribute can be checked to find the subclass type. 8d) • Create a relational table for the superclass that includes all the components (attributes, domains, constraints) of all the subclasses. • Include an additional Boolean flag for each kind of subclass in the model. • This flag is set true for the subclass. • Good for Overlapping.

• If a subclass cannot have a shared role with any other subclass, the specialization is called disjoint

• If a subclasses can have shared roles with any other subclass,
the specialization is called overlapping,