

SAD II

Pretest for Chapter 14

Normalize the following document.
Show UNF, 1NF, 2NF and 3NF

1. CUSTOMER ORDER

ORDER NO: 61384
CUSTOMER NO: 1273
CUSTOMER NAME: CONTEMPORARY DESIGNS
CUSTOMER ADDRESS: 123 OAK ST.
CITY STATE ZIP: AUSTIN, TX 28384

ORDER DATE: 9/24/2000

| PRODUCT NO | DESCRIPTION | QUANTITY ORDERED | UNIT PRICE | EXTENDED PRICE |
|---------------|-------------|---------------------|---------------|-------------------|
| M128 | BOOKCASE | 4 | 200.00 | 800.00 |
| B381 | CABINET | 2 | 150.00 | 300.00 |
| R210 | TABLE | 1 | 500.00 | 500.00 |
| | | | TOTAL | 1600.00 |

Given the following DBDL. . Produce an ERD showing key and non-key attributes. Also produce an event analysis (referential integrity).

ORDER(Order #, *Cust Number*)
FK Cust Number → CUSTOMER

CUSTOMER(Cust Number, Cust Name, State, City, Zip, Discount %)

ORDERPRODUCT(Order #, Product No, Qty Order)
FK Order # → ORDER
FK Product No → PRODUCT

PRODUCT(Product No, Prod Desc)

Draw your ERD here.

Data Normalization

Data Normalization is similar to drawing ERD's and converting them into relational tables, except that in the Normalization process, you start with the attributes and work back to the entities to set up your tables. When working with actual systems, you will generally find the attributes that you need to start on one of the sample forms that you acquired from the person who wants you to build the database.

Normalization should be used in addition to drawing ERD's to insure that your database is set up correctly. You should do both and then check them against each other to make sure that the results are the same. They should be. If your database is not set up correctly, you will only run into trouble later on.

There are 4 steps to creating a "normalized" database:

- ❶ Represent the user view(s) by listing all of the data attributes in a logical order. Name the table and identify the primary key(s). Follow the DBDL rules. (See Database Design Language handout)
- ❷ Place the tables into **First Normal Form**. A table is in First Normal Form (1NF) if no attributes form repeating groups. You remove the repeating groups and use them to form another table. You must *link* the two tables, by posting the key from the original table into the new table. Once this is done you should look at the tables that you have and check to see if these tables are in First Normal Form. If not, remove the repeating groups according to the rules at the beginning of this item.
- ❸ Place the tables into **Second Normal Form**. A table is in Second Normal Form (2NF) when all nonkey attributes are functionally dependent on the entire key. 2NF is targeted at tables in which the records (entities) are identified by a concatenated key. This can be achieved by removing partial key dependency and placing it in a new table with its corresponding key. You must also leave that key in the original table.
- ❹ Place the tables into **Third Normal Form**. A relational table is in Third Normal Form (3NF) when it is in 2NF and no transitive dependencies exist. A transitive dependency occurs when a nonkey attribute is fully dependent on another nonkey attribute. This relationship is removed into its own table, leaving the new table's key in the original table as a *foreign key*. Go back to see if you have any 2NF problems, if so correct them before you can re-evaluate your tables for 3NF problems.

Database Design Language (DBDL)

Database Design Language (DBDL)

- Capitalize the relation name
- Put attributes in parenthesis
- Underline primary keys
- Italicize foreign keys
- Attributes that are allowed to be null are followed by an asterisk
- Foreign keys are identified by the letters FK followed by the attribute(s) that comprise the foreign key
- Foreign keys are followed by an arrow pointing to the relation identified by the foreign key

DBDL Example

DEPT (Deptnumb, Deptname)

EMPLOYEE (Empnumb, Empname, Empaddr*, Ssnnumb, *Deptnumb*, ...)
FK Deptnumb --> DEPT

Event Analysis

Parent to Child Integrity Rules:

Parent Key is part of the Primary Key in the Child Entity:

Delete: Cascade

A deletion of a record in the Parent entity must be automatically followed by the deletion of matching records in the Child entity.

Delete: Restrict

A deletion of a record in the Parent entity must be disallowed.

Insert: Cascade

An insertion of a record in the Parent entity must be automatically followed by the insertion of matching records in the Child entity. (This is used if the cardinality of the Child entity is 1 or M)

Insert: No Restrictions

An insertion of a record in the Parent entity will not insert a matching record in the Child entity. (This is used if the cardinality of the Child entity is 0 or M)

Parent Key is a Foreign Key in the Child Entity:

Delete: Cascade

A deletion of a record in the Parent entity must be automatically followed by the deletion of matching records in the Child entity. (Most Common)

Delete: Restrict

A deletion of a record in the Parent entity must be disallowed.

Delete: Set Null or Default

A deletion of a record in the Parent entity will be followed by setting any matching foreign keys in the Child entity, to NULL or a Default value.

Insert: Cascade

An insertion of a record in the Parent entity must be automatically followed by the insertion of matching records in the Child entity. (This is used if the cardinality of the Child entity is 1 or M, rare.)

Insert: No Restrictions

An insertions of a record in the Parent entity will not insert a matching record in the Child entity. (This is used if the cardinality of the Child entity is 0 or M)

Child to Parent Integrity Rules:

Parent Key is part of the Primary Key in the Child Entity:

Delete: Cascade

A deletion of a record in the Child entity must be automatically followed by the deletion of matching record in the Parent entity. (This is used if the cardinality of the Child is 1 or M, and this is the last Child record for the Parent. The last Order Detail record for an Order. Sometimes this is called the Salmon delete, because it deletes upstream.)

Delete: No Restrictions

A deletion of a record in the Child entity will have no effect on the Parent entity. (This is used if the cardinality of the Child is 0 or M)

Insert: Restrict

An insertion of a record in the Child entity must be disallowed until a matching record is inserted in the Parent entity.

Parent Key is a Foreign Key in the Child Entity:

Delete: Cascade

A deletion of a record in the Child entity must be automatically followed by the deletion of matching record in the Parent entity. (This is used if the cardinality of the Child is 1 or M. Rare)

Delete: No Restrictions

When a child Entity is deleted, there is no effect on the Parent entity. (This is used if the cardinality of the Child is 0 or M)

Insert: Restrict

An insertion of a record in the Child entity must be disallowed until the matching record is inserted in the Parent entity. (Most Common)

Insert: Set Null or Default

An insertion of a record in the Child entity will be followed by setting the foreign keys in the Child entity, to NULL. or a Default value

Update: Restrict

An update of a record in the Child entity must be disallowed until the matching record is inserted in the Parent entity. (Most Common)

Update: Set Null or Default

An update of a record in the Child entity will be followed by setting the foreign keys in the Child entity, to NULL. or a Default value