# Normalization Packet

# *Table of Contents*

# Steps for 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

# Exercise #1 - Ticktown

Ticktown is famous for the fact that each person in town owns at least one dog. The number of practicing vets that treat dogs in Ticktown is astronomical compared to any other place on earth. As a matter of fact, Ticktown's nickname is "Dog Heaven".

Each vet has a name, address, and phone number as well as a license number. Each vet's license number is different from any other vet's license number. Each Ticktown vet has a workload which indicates the number of dogs the vet has treated in a year.

Each person is identified by a name and lives in a house with a unique address.
More than one person can live in the same house. Each dog is owned by one person.
However, each person can own more than one dog.

People in Ticktown never get their dogs mixed up because each dog is distinguishable from every other dog by its dog tag number. Each dog has a name. Each dog is treated by only one vet.

Normalize this problem.

# Exercise #2 – Lee's Video Store

Lee is interested in movies and wants to keep information on movies, stars, and directors in a database. The example contains the following requirements:

1. For each director, list his or her number, name, the year he or she was born, and, if he or she is deceased, the year of death.

2. For each movie, list its number, title, the year the movie was made, its type, the number and name of its director, the critics' rating, the MPAA rating, the number of awards the movie was nominated for, the number it won and the names of all the stars that appeared in it.

3. For each movie star, list his or her number, name, birthplace, the year he or she was born, and, if he or she is deceased, the year of death.

For this project, we will add the following requirements to the above list:

Requirement 1:

Each video tape purchased by Lee's store is assigned a unique number. For each tape, Lee wants to store the number; the date the tape was purchased; the number, name, and address of the supplier from which the tape was purchased; and the number and title of the movie that is on the tape. Although Lee will usually only have one tape for any given movie, he will often purchase several tapes of the movies that he feels are especially popular. If Lee purchases more than 1 tape, he will purchase them all from the same supplier.

Requirement 2:

For each customer, store the customer's number, name, street address, city, state and zip. In addition, we must be able to determine precisely which tapes the customer has rented along with the date and time at which they were rented as well as the date and time that they are due to be returned.

Normalize this problem

# Exercise #3 – Ashe & Macomb

It was about time for the small accounting firm of Ashe & Macomb to computerize their record keeping procedures. Business had grown by "leaps and bounds" first half of the new year and the two partners couldn't keep up with all the new business because of the outdated record keeping procedures they used.

As one example of how outdated things were at A & M, they produced ledger reports by hand for their customers! Now that business had accelerated, the number of mistakes they were finding was increasing. They had to do something.

One of the ledger reports for a new (and very large) customer, Denton Packaging, is illustrated. The information for each row and column had to be input by hand the old way to create the report. Each transaction had to hand sorted into the correct row. It was a nightmare!

A ledger report consists of data about various types of financial transactions, in the case of Denton Packaging, for each department within each of its divisions.

A ledger account is important to ledger report because it helps to uniquely identify each financial transaction. Each ledger account is different from all others because each ledger account has a unique code number. Each ledger account also has a description and is a debit or credit.

Each transaction comes from a department of Denton Packaging and consists of a date, ledger account, amount and comment. Each transaction can be uniquely identified by knowing its ledger account, date and department.

Each division can be identified by a division number and has a name. Each department can be identified by knowing its department number and its division. Each department also has a name.

Ashe & Macomb Ledger Report for Denton Packaging

| Date | Amt | Comment | Code | Description | D C | No | Name | Div No | Name |
|---|---|---|---|---|---|---|---|---|---|
| 01/17/02 | 150 | Office Party | 2001 | Petty Cash | D | 170 | Accounting | 100 | Comsumer Products |
| 01/18/02 | 1500 | Electronic Parts | 1020 | Invoices Pymt Rcv | C | 150 | Inv Mgmt | 200 | Commercial Products |
| 01/18/02 | 2900 | CDs and Tapes | 1015 | Credit Card Sales | C | 130 | Marketing | 100 | Consumer Products |
| 01/21/02 | 5000 | Payroll | 2053 | Payroll Expense | D | 170 | Accounting | 100 | Consumer Products |
| 01/24/02 | 75 | Office Meeting | 2001 | Petty Cash | D | 130 | Marketing | 100 | Consumer Products |

# Exercise #4 – MSTC

MSTC needs to store information about Departments, Faculty, Students and Courses. The example contains the following requirements.

1. For a department, store the department number and name.

2. For a faculty member, store the number and name.

3. For a student, store the number and name.

4. For a course, store the code and title.

5. For each student, store the grade earned in each of the courses he or she has taken.

6. For each faculty member and each course taught by the faculty member, store the number of times he or she has taught the course. A course can be taught by more than one faculty member.

7. Each department may contain any number of faculty members, but each faculty member is assigned to exactly one department.

8. Each faculty member may advise any number of students, but each student is advised by, at most, one faculty member.

9. The system must be able to efficiently retrieve a department, faculty member, student, or course based on the department number, the faculty number, the student number or the course code respectively. A grade does not need to be retrieved directly, neither does the number of times a faculty member has taught a course.

10. A faculty member cannot be added to the database unless the department to which the faculty member is assigned is known and already stored in the database. Once a faculty member is stored, the department to which the faculty member is assigned cannot be changed.

11. A student need not have an advisor to be stored. Further, once a student is stored, it must still be possible to change the student's advisor.

# Exercise #5 – Vet

Dr. Good is a Veterinarian who currently keeps all of his office records in a paper filing system. He would like it computerized because:

- record keeping is tedious and cumbersome
- records are not backed up and could all be lost in an event of a physical disaster (fire, flood, etc.)
- information is not readily available. It can take days to compile a report, for example, about how many dogs were treated for Lymes Disease this year.
- Calculating customer bills is difficult

The records are kept in the following system at this time. Find a way to reorganize it into a relational database.

Normalize this problem.

---

The doctor keeps a list of medications that are available in a spreadsheet. It looks like this:

| Medication Name | Dose/5 Pounds | Price/Dose |
|---|---|---|
| Acetimophenabil | 20 mL | 1.1 |
| Echerschnial | 12.5 grams | 0.25 |
| Ehisyvia | 1 tsp | 0.59 |
| Empernalia | 1/4 ounce | 0.67 |
| Hemperstau | 1/2 tsp | 2.25 |
| Rakor | 1/2 ounce | 0.12 |
| Raxora | 1.6 ounces | 0.15 |

---

The doctor posts in this office lobby a list of the types of animals that he sees. It includes: dogs, cats, mini-pigs, and ferrets. He may expand his practice in the future to include other types of animals.

---

# Customer Form

He has new customers fill out the following form:

NEW CUSTOMER FORM

Today's Date: _____/_____/_____

First Name: _____ Last Name: _____

Street: _____ City: _____

State: ___ ZIP: _____ Phone: _____

Please provide the following information for each animal that you plan to bring into our clinic. If you own more than one animal, please ask the receptionist for an additional form. Do not complete the top part again, simply staple the pages together.

Pet Name: _____

Type of Animal: Dog Cat Ferret Pig
Breed: _____ (ex: Cocker Spaniel)

Date of Birth: _____

Sex: M F

Color(s): _____

Neutered/Spayed? Yes No
Weight: _____

Current Vaccinations: _____

Other Comments: _____
_____

# Visit Form

Each time a pet visits, the clerk fills out the following form and gives it to the doctor to complete during the visit. The clerk fills in the upper part and the doctor fills in the bottom part. It is then filed in the drawer by customer with the original new customer form. There can be more than one visit type per visit.

Pet Name: _____

Owner Name: _____

Animal Date of Birth: ____/____/____

Current Weight: _____

Reported Reason for Visit: _____

Date of Visit: ____/____/____

Visit Type:_____
(Routine Vaccination, Physical, Ailment, Surgery, etc.)

Diagnosis/Comments:
_____
_____
_____
_____

Follow up Appointment Necessary? Yes No

Prescriptions:
Medication Name: _____ Dosage: _____

Medication Name: _____ Dosage: _____

Medication Name: _____ Dosage: _____

Medication Name: _____ Dosage: _____

Total Bill: _____

# Fees

The doctor has a chart of how much different types of visits cost to help in calculating the fees to be charged:

| | |
|---|---|
| Routine Vaccination | $5.00 |
| Physical | $15.00 |
| Ailment | $30.00 |
| Surgery | |
| Clean Teeth | $25.00 |
| Minor | $75.00 |
| Major | $175.00 |
| Specialty | varies |

# Pretest #1: ERD & Normalization

Step 1: Entity Discovery. (1 point)
Step 2: Relational Matrix. (1 point)
Step 3: Produce a Context ERD. (1 point)
Step 4: Produce a Key-Base ERD. (1 point)
Step 5: Produce a Fully Attributed ERD. (1 point)
Step 6: Produce an UNF for the PICKING LIST DISPLAY. (1 point)
Step 7: Produce a 1NF for the PICKING LIST DISPLAY. (1 point)
Step 8: Produce a 2NF for the PICKING LIST DISPLAY. (1 point)
Step 9: Produce a 3NF for the PICKING LIST DISPLAY. (1 point)
Step 10: List the entities in DBDL with foreign keys. (1 point)

PICKING LIST DISPLAY:

A picking list is a list of the parts needed to assemble a product, in this case a front bicycle wheel. There is a many to many relationship between Item and Part.

Hint: Use Item number as your key for UNF.

PICKING LIST

INVENTORY ITEM NUMBER: W150
INVENTORY ITEM DESCRIPTION: FRONT WHEEL ASSEMBLE A

| PART # | DESCRIPTION | QTY |
|---|---|---|
| M300 | CHAMPION RIM | 1 |
| U600 | SPECIALIZED HUB | 1 |
| S101 | CHAMPION SPOKES | 36 |
| T003 | SPEC TOURING TIRE | 1 |
| E102 | FUJI TUBE | 1 |

# Pretest #2: ERD & Normalization

Step 1: Entity Discovery. (1 point)
Step 2: Relational Matrix. (1 point)
Step 3: Produce a Context ERD. (1 point)
Step 4: Produce a Key-Base ERD. (1 point)
Step 5: Produce a Fully Attributed ERD. (1 point)
Step 6: Produce an UNF for the PICKING LIST DISPLAY. (1 point)
Step 7: Produce a 1NF for the PICKING LIST DISPLAY. (1 point)
Step 8: Produce a 2NF for the PICKING LIST DISPLAY. (1 point)
Step 9: Produce a 3NF for the PICKING LIST DISPLAY. (1 point)
Step 10: List the entities in DBDL with foreign keys. (1 point)

1. **CUSTOMER ORDER**

ORDER NO: 61384 ORDER DATE: 9/24/94
CUSTOMER NO: 1273
CUSTOMER NAME: CONTEMPORARY DESIGNS
CUSTOMER ADDRESS: 123 OAK ST.
CITY STATE ZIP: AUSTIN, TX 28384

| PRODUCT NO | DESCRIPTION | QUANTITY ORDERED | UNIT PRICE | TOTAL 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 |

2. **CUSTOMER DISPLAY**

CUSTOMER NO: 1273
NAME: Contemporary Designs
ADDRESS: 123 OAK ST
CITY-STATE-ZIP: AUSTIN, TX 28384
YTD-PURCHASES: $47,284.29
CREDIT LIMIT: $10,000
OUTSTANDING BALANCE: $4628.34

3. **PRODUCT DISPLAY**

PRODUCT NO: M128
DESCRIPTION: BOOKCASE
LOCATION: B
QUANTITY ON HAND: 10
REORDER POINT: 20
UNIT PRICE: $200.00

# Pretest #3: ERD & Normalization

Step 1: Entity Discovery. (1 point)
Step 2: Relational Matrix. (1 point)
Step 3: Produce a Context ERD. (1 point)
Step 4: Produce a Key-Base ERD. (1 point)
Step 5: Produce a Fully Attributed ERD. (1 point)
Step 6: Produce an UNF for the PICKING LIST DISPLAY. (1 point)
Step 7: Produce a 1NF for the PICKING LIST DISPLAY. (1 point)
Step 8: Produce a 2NF for the PICKING LIST DISPLAY. (1 point)
Step 9: Produce a 3NF for the PICKING LIST DISPLAY. (1 point)
Step 10: List the entities in DBDL with foreign keys. (1 point)

**1. Patient Display**

| | |
|---|---|
| PATIENT-NO: | 12345 |
| PATIENT-NAME: | Baker, Mary A. |
| PATIENT-ADDRESS: | 300 Oak St. |
| CITY-STATE-ZIP: | Mountain View, CO 80638 |
| SEX: | F |
| SOC-SECURITY-NO: | 376-38-0458 |
| LOCATION: | 328B |
| EXTENSION: | 623 |
| DATE-ADMITTED: | 07-14-2002 |
| FINANCIAL-STATUS: | Prudential |
| DISCHARGE-DATE: | |

Notes:
Location is the room and bed location to which the patient is assigned. The first 3 digits are numeric and indicate the room number; the last character is alphabetic and indicates the bed in the room. All locations are unique.

Extension is the patient's phone extension. Each location has an extension.

Financial-Status is the patients primary source of payments for hospital service.

2. **Physician Display**

| | |
|---|---|
| PHYSICIAN-NO: | 4321 |
| PHY-NAME: | M. D. Thayer |
| PHY-PHONE: | 608-5585 |
| SPECIALTY: | Pediatrics |

Notes:
Specialty identifies the physician's field of specialization.

3. **Physician-Patient Display**

PHYSICIAN-NO: 4024
PHY-NAME: Dunn, A. J.

| PATIENT_NO | PATIENT-NAME | LOCATION | DATE-ADMITTED |
|---|---|---|---|
| 12870 | Gonzalez, P.T | 103A | 01-12-2003 |
| 23819 | Thomas, Marie | 214C | 02-02-2003 |
| 61431 | Cuadra, L. R. | 281B | 02-07-2003 |

Notes: Patients can be treated by more than one doctor.