Eddie Gurnee
COSC 571
1/28/15

Homework 1

Chapter 1:

1.3) List six major steps that you would take in setting up a database for a particular enterprise:

Response:
1. Requirement Analysis
2. Conceptual Design
3. DBMS Selection
4. Logical Design
5. Physical Design
6. Create and initialize the database

1.8) List four significant differences between a file-processing system and a DBMS:

Response:
1. Databases reduce data redundancy and inconsistency, redundancy that exists in controlled redundancies.
2. File-processing systems need specific programs to open each individual file, where databases can be accessed entirely with one application through SQL.
3. Databases allow for the relations between data to be expressed, file systems do not inherently have the ability to show relationships.
4. Data in file systems is stored in multiple files and formats, databases instead allow for data to be stored in a centralized location.

1.9) Explain the concept of physical data independence, and its importance in database systems.

Response: Physical data independence is the concept that the implementation of lowest level of data storage (the physical layer) is not needed to be understood or considered at higher levels. This abstraction allows for database admins to simply worry about the data instead of the potential complexity of the physical layer.

1.11) List at least two reasons why database systems support data manipulation using a declarative query language such as SQL, instead of just providing a library of C or C++ functions to carry out data manipulation.

Response:
1. DMLs remove any ambiguity that exists from a specific language implementation of database manipulation, by prescribing a very specific procedure and set of rules to follow for anything that would use the DML.
2. Declarative query languages are based on relational algebra and calculus; as such, someone attempting to utilize such data interactions requires a basic understanding of set theory and relations instead of a specific language's idiosyncrasies.

1.12) Explain what problems are caused by the design of the table in Figure 1.4.

Response: With all of the data being stored in one table, there exists a large number of redundancies; besides the fact that redundant data wastes space, updating the data now needs to be done in multiple locations instead of updating in one location and allowing relations to do the work. Additionally, some information cannot be represented; for instance, new departments cannot simply be created without first an instructor being assigned to it.

Chapter 2:

2.6) Consider the following expressions, which use the result of a relational algebra operation as the input to another operation. For each expression, explain in words what the expression does.

a. $\sigma_{year} \geq 2009 \text{ (takes)} \bowtie \text{student}$

b. $\sigma_{year} \geq 2009 \text{ (takes} \bowtie \text{student)}$

c. $\Pi_{ID,\ name,\ course\_id} \text{ (student} \bowtie \text{takes)}$

Response:

a) Looking at the data of each instance of a class that was taken in the year 2009 and beyond, match that data with the data that each student that took said class.

b) After matching the data of each instance of a class that was taken with the data of the student who took it, look at that data that from the year 2009 and beyond.

c) After matching the data of each student with the data of each class they have taken, display only the Student ID, Student Name, and Course ID.

2.9) Consider the bank database of Figure 2.15.

a. What are the appropriate primary keys?

b. Given your choice of primary keys, identify appropriate foreign keys.

Response:

branch (branch_name, branch_city, assets)
customer (customer_name, customer_street, customer_city)
loan (loan_number, *branch_name*, amount)
borrower (*customer_name*, *loan_number*)
account (account_number, *branch_name*, balance)
depositor (*customer_name*, *account_number*)
-Bank database from Figure 2.15

2.10) Consider the advisor relation shown in Figure 2.8, with s_id as the primary key of advisor. Suppose a student can have more than one advisor. Then, would s_id still be a primary key of the advisor relation? If not, what should the primary key of advisor be?

Response: No, s_id would no longer be the primary key of the advisor relation, as student's could have more than one advisor, there is no guarantee that the s_id would remain unique. The primary key of the advisor relationship would now have to become both the s_id and i_id, student and instructor id respectively.

2.13) Consider the bank database of Figure 2.15. Give an expression in the relational algebra for each of the following queries:

a. Find all loan numbers with a loan value greater than $10,000.

b. Find the names of all depositors who have an account with a value greater than $6,000.

c. Find the names of all depositors who have an account with a value greater than $6,000 at the “Uptown” branch.

Response:

a) $\Pi_{loan\_number} (\sigma_{amount>10000} \text{ loan})$

b) $\Pi_{customer\_name} (\sigma_{balance>6000} \text{ (depositor} \bowtie \text{account))}$

c) $\Pi_{customer\_name} (\sigma_{balance>6000,branch\_name='Uptown'} \text{ (depositor} \bowtie \text{account))}$

2.16) Differentiate between the following

a) Superkey vs. candidate key

b) Primary key vs. foreign key

c) Schema vs. instance

d) Procedural vs. non-procedural query languages
e) Selection vs. projection operations

Response:
a) A superkey of a relation is a set of one or more attributes whose values are guaranteed to identify tuples in the relation uniquely, where a candidate key is a minimal superkey, it is the smallest combination of attributes needed to make a relation unique.
b) A primary key is a candidate key chosen to identify tuples of a relation, while a foreign key is the primary key of a tuple used as an attribute in another tuple to indicate the existence of a relationship.
c) A schema of a relation or database refers to its logical design, while an instance of either refers to its contents at a point in time.
d) In a procedural query language, the user instructs the system to perform a sequence of operations on the database to compute the desired result, while in a nonprocedural language the user describes the desired information without giving a specific procedure for obtaining that information.
e) Selection operations retrieve whole tuples of a relation while a projection includes only the attributes that have been specified by the query.

<u>Chapter 3:</u>

3.25) DDL/DML written exercise.

a. Write SQL DDL statements corresponding to the schema in Fig. 3.18 (Insurance database) in the textbook. Make any reasonable assumptions about the data types and be sure to declare primary and foreign keys.

b. Write SQL DDL/DML statements to do the following:

i. Alter any one table to include a new, meaningful attribute.

ii. Insert one tuple each into each of the tables. Make sure the data follows all integrity constraints.

iii. Delete one tuple from any table (do not delete all tuples).

iv. Update any one tuple from any table.

```
CREATE DATABASE Insurance;

a)
CREATE TABLE person
(
driver_id int NOT NULL,
name varchar(20),
address varchar(20),
primary key (driver_id)
);

CREATE TABLE car
(
license varchar(8) NOT NULL,
model varchar(20) NOT NULL,
year numeric(4,0),
primary key (license)
);

CREATE TABLE accident
(
report_number int NOT NULL,
```

```
date numeric(6, 0) NOT NULL,
location varchar(20) NOT NULL,
primary key(report_number)
);

CREATE TABLE owns
(
driver_id int NOT NULL,
license varchar(8) NOT NULL,
primary key(driver_id, license),
foreign key (driver_id) references person,
foreign key (license) references car
);

CREATE TABLE participated
(
report_number int NOT NULL,
license varchar(8) NOT NULL,
driver_id int NOT NULL,
damage_amount real NOT NULL,
primary key(report_number, license),
foreign key (report_number) references accident,
foreign key (license) references car,
foreign key (driver_id) references person
);
```

b)

i.

```
ALTER TABLE person
ADD phone_number int(10);
```

ii.

```
INSERT INTO person
(driver_id) VALUES (101);

INSERT INTO car
VALUES (‘BKZ 432X’, ‘Pilot’,  2000);

INSERT INTO accident
VALUES (2, 102790, ‘The Moon’);

INSERT INTO owns
VALUES (123, ‘BKZ 432X’);

INSERT INTO participated
VALUES (2, ‘BKZ 432X’, 101, 54.4);
```

iii.

```
DELETE FROM person
WHERE driver_id = 101;
```

iv.

```
UPDATE person
SET name='carlos'
WHERE driver_id=101;
```