

Data Management and Database Design

INFO 6210

Fall 2016

Assignment 6

Submission Date – Nov/13/2016 Sunday

Student Name: Malick Fairoz Sayeed Abuthahir

NUID: 001235450

Program: MS in Information Systems

Professor Name: Yusuf Ozbek

College: College Of Engineering

University: Northeastern University

PART – 1

Chapter 5

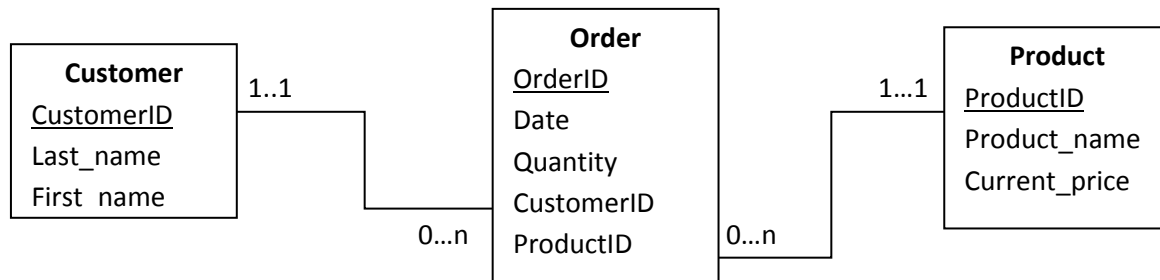
- 1) CREATE TABLE BOOKS(Select BookId,BookName from Books);
- 2) UPDATE BOOKS SET BookId=1001,BookName=' One Hundred Years of Solitude'
where BookId=1001
- 3) update CDs set CDQuantity = CDQuantity +3;
- 4) update CDs set CDQuantity = CDQuantity +3 WHERE CDName = 'Mule Variations';
- 5) delete from CDs where CDID=1;

Chapter 6

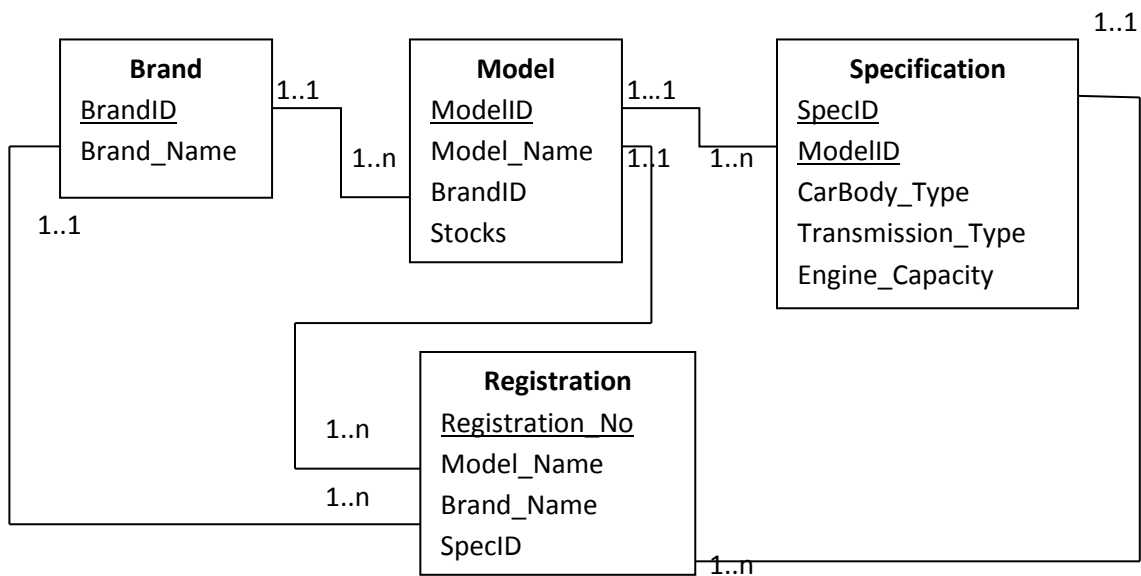
- 1) CREATE SCHEMA `newdb` DEFAULT CHARACTER SET latin1 COLLATE latin1_general_ci ;
- 2) CREATE TABLE `dvdrentals`.`Bikes` (
`BikeID` INT NOT NULL AUTO_INCREMENT,
`BikeName` VARCHAR(40) NOT NULL,
PRIMARY KEY (`BikeID`),
UNIQUE INDEX `BikeID_UNIQUE` (`BikeID` ASC),
UNIQUE INDEX `BikeName_UNIQUE` (`BikeName` ASC));
CREATE TRIGGER tablelimit
BEFORE INSERT
ON Bikes
FOR EACH ROW
BEGIN
SELECT COUNT(*) INTO @cnt FROM Bikes;
IF @cnt >= 200 THEN
CALL sth(); -- raise an error
END IF;
END
\$\$
- 3) ALTER TABLE `dvdrentals`.`modeltrains` ADD UNIQUE INDEX `UN_1` (`ModelName` ASC);
- 4) ALTER TABLE `dvdrentals`.`modeltrains` DROP INDEX `UN_1` ;
- 5) Show tables;

PART – 2

1) The given table cab be developed as following



2) Car sales yard tables can be described as below



PART – 3

- 1) Select * from edition;
- 2) Select ISBN from edition where FORMAT = 'hardcover';
- 3) Select ISBN from edition where FORMAT = 'hardcover' group by Format order by list_price;
- 4) The value of “select 3 + 4 * 5” is “23” and the value of “select (3 + 4) * 5” is “35” so the expression (3 + 4) * 5 yields greater value.
- 5) select sum(LIST_PRICE) from edition group by PUBLISHER_ID;
- 6) select * from subject as s inner join publication as p on s.SUBJECT_ID=p.SUBJECT_ID;
- 7) select * from subject as s left outer join publication as p on s.SUBJECT_ID=p.SUBJECT_ID;
- 8) The two different ways can be written in IN and EXISTS clause
 - a. select * from subject where subject_id in (select subject_id from publication);
 - b. select * from subject where exists (select from publication where subject.subject_id=publication.subject_id);

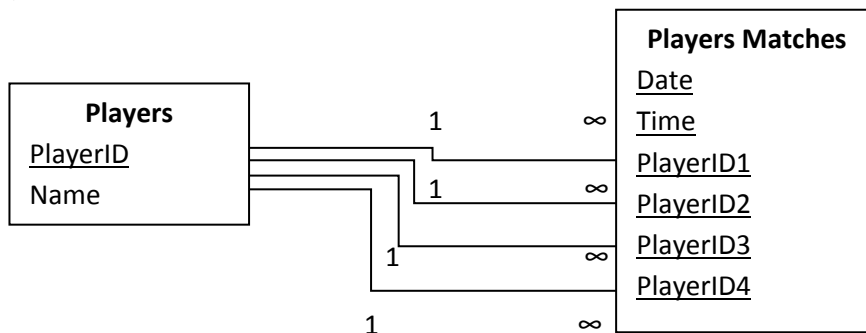
PART – 4

Chapter 9

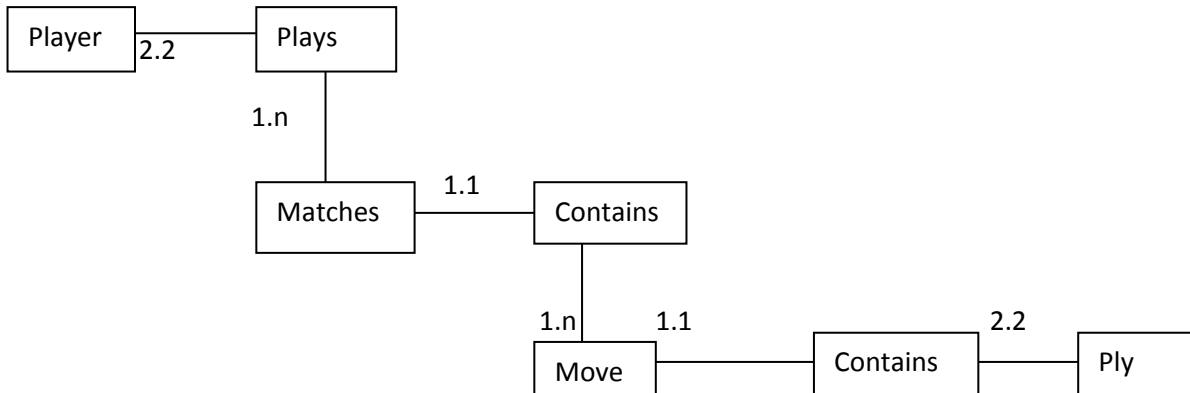
- 1) The ER diagram of Parcheesi is



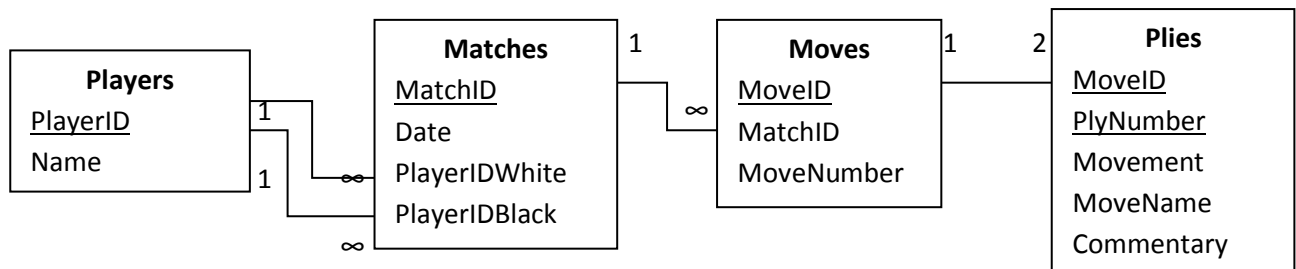
- 2) Relational model is



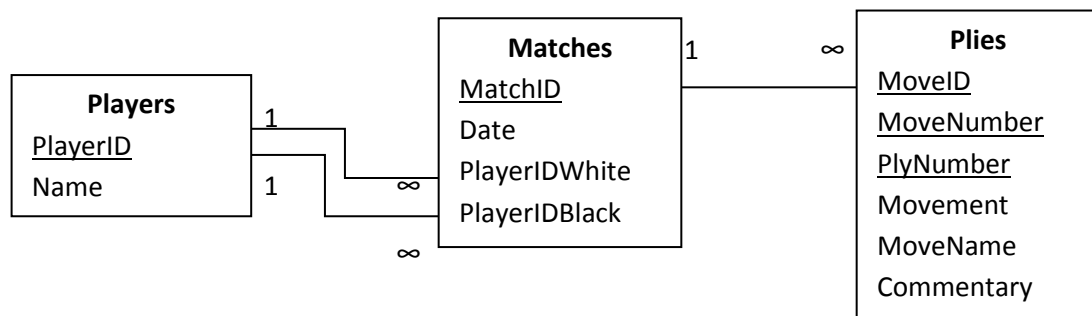
3)



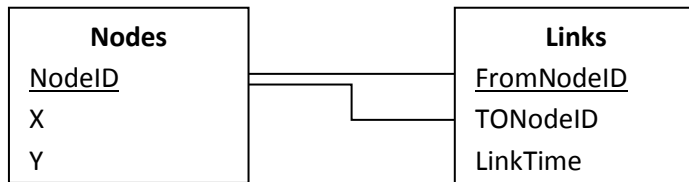
4) Relational Model



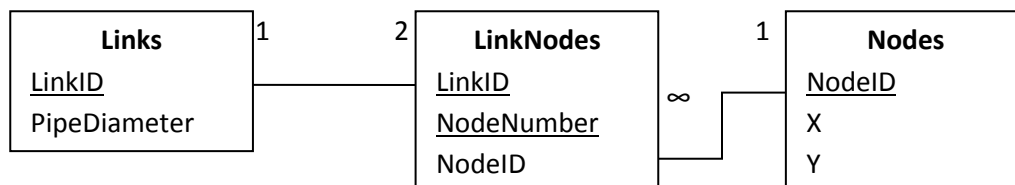
5) Chess model without Move table



6)

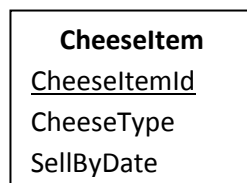


Consider the above two tables for modeling the network of pipes were FromNodeID and ToNodeID imply a direction for the link. This represent the link record connecting node 1 and node 2 also represent link connecting record from node2 to node1. For normalization both fromnodeid and tonodeid should store same kind of data.

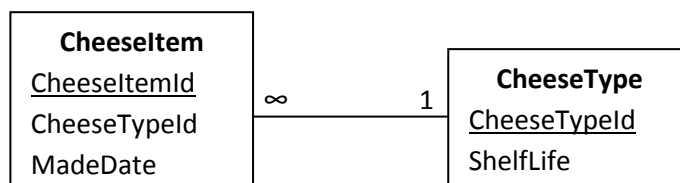


This is also same normalization design for a undirected network, the only difference is in undirected network the nodenumber field has a index with two nodes whereas in directed network it has separate node which is from and to.

7) The model for Cheese Item is shown below. It holds other data like quantity of cheese , types and dates.



8)



This model represents more consistency since it has common shelf life for each cheese type.

Chapter 10

- 1) This customer table has lot of problem which are listed below,
 - a. In this table columns has multi value entries which shows it does not even in first normal form.
 - b. Client wants to get the state value from zip code but why not get the value of city? Or for even more better we can have separate table for city, state, zip code, country.
 - c. There are two phone number fields but there is not differentiate between both like which is mobile number or office number or home number, also there is no indication when these phone numbers has available timing.
 - d. If the customer has more than two contact numbers then there is not option to enter the details
 - e. In the address field there is no separate column location for data. It can be taken as columns like street 1 and street 2. So that we can have more clear perspective view.
 - f. The customers interest details can be updated in a separate look up table and can be used with referential constrains.
 - g. In Future we can add other more details like contact Email Id.

2)

Since it is a new opening store, we should wonder if they will grow more in future. And should design accordingly. Blue ray is a brand bnw technology and might increase the demand for rentals in the stores. For this database we should the at very high loads to test that it can run on ever increasing data.

On the other hand a well developed store probably might noy experience high growth in the short future because it is not a new technology. So need to test the application and not required high load above the present level.

3)

This table is hyper-normalized. Though you can break a street address into name, number, prefix, and so forth, there are very few applications where that is necessary. If you will only ever need to use the address information to send mail to someone, you can combine all of this information in a single Street field. You can even include the apartment or suite number. If you're only going to use the ZIP Code to write addresses, there's no need to use separate fields. Here's the new list of fields:

- CustomerId
- Street
- City
- State
- Zip

4)

In this model, the Phones table is fairly unconstrained because it allows a person to have any number of any type of phone number. All of the fields are required. Some other validations that you could build into the database include:

Field	Constrain	Implementation
PersonId	Exists	Foreign key match to Persons.PersonID
Type	Enumerated value	Foreign key match to new phone types table
Number	Format	Let the database verify that the value has format #

In the Persons table, every field except MiddleName should be required. The table can implement the following constraints:

Field	Constraint	Implementation
State	Enumerated value	Foreign key match to new states table
Zip	Format	Let the database verify that the value has format #

Finally, you could check that the Date fields in the CourseResults and ProjectResults tables come after the corresponding student's enrollment date.