**Database Assignment 1** 10/10/2017

**Aim**

To design a database for a given case study ‘Games Galore’ shop. To make the appropriate design decisions when undertaking logical and physical data modelling of Games Galore information system that can support all their retail activities in their shop.

**Supplier**

In the supplier entity the decision what made to break contact details into two attributes first name and phone number. The supplier name, email, and phone number all are set to UNIQUE and NOT NULL. Also, a validation constraint was put on supplier email to ensure that an @ symbol is inserted (companyEmail Like '%@%'). A phone contact first name is optional for the system so it is left an NULL.

**Sale Rep**

Thesale rep entity is required for making orders between customers and suppliers. In the sales rep entity the decision was made to break the rep name into first name and second name. This was done to prevent a compound attribute and make each attribute into it most atomic state. A validation constraint was put on repEmail which will only let reps enter emails if they put in an @ symbol. (repEmail Like ‘%@%’). Sales rep email will also be set as unique so no sales rep can enter the same email address.

**Supplier & Sales Rep**

The relationship between sales rep and supplier will be many to many because many suppliers can have many sales reps and many sales reps can work for many suppliers. The primary key will be a set as the supplier.

**Supplier & Accessory**

There will be a many to many relationships between suppliers and accessory. Many different suppliers can supply many accessory’s. In the physical view an accessory id will be added to identify which accessory will be supplied by which supplier.

**Console**

The console entity will consist of one attribute called console name. Console name will be set as the primary key because it is unique.

**Supplier & Console**

There will be a many to many relationships between suppliers and console. Many suppliers can supply many consoles. A console ID will be added in the physical layer bridge table because many suppliers can have the same console. The console ID will uniquely identify where the console is supplied from.

**Game**

Game entity will have the following two attributes game name and release date. The decision was made to make skill level out and age group their own entities. This was done to prevent anomalies from occurring. Game has the foreign key of age group.

**Skill level**

A decision was made to make skill level its own entity to prevent anomalies from occurring in game entity. The decision was made to have the relationship one to many because a game can only be of one on skill level. e.g. advanced or beginners. it is also non-identifying because the game identity doesn’t rely on it to uniquely identify itself.

**ageGroup**

Age group is its own entity. This decision was made to prevent repeating groups in game entity. It’s relationship with game is one to many because one game can only have one age group and one age can be of many games. it is also non-identifying because the game identity doesn’t rely on it to uniquely identify itself.

**Game & Console (Console usable bridge table)**

The relationship between game and console will be many to many because one game can be used on many consoles and many consoles can have many games. A composite key will be set up between game name and console name. In the bridge table an additional attribute called usable console will be added to indicate if a game can be used on that console.

**Supplier & Game (Supplier game stock bridge table)**

There will be a many to many relationships between game and supplier. Many different suppliers can supply a game and many different games can be supplied by many suppliers. The same game can be supplied by many suppliers so gameID is added in physical view to identify where the game is supplied from.

**Accessory/Accessory & Console Compatibility bridge table**

Accessory entity will have two attributes name and type. Type will have a constraint of either generic or specific. Its relationship with console will be many to many on the bridge table on the physical view the attribute compatible will be added so user can answer yes or no, either it is compatible with that console or not.

**Item**

ItemID is the best candidate for primary key because all the rest of the attributes in the entity are dependent on it. Item entity will have the following attributes all of which will be set as NOT NULL unitInStock , markUpPrice, decryption and price. Item type id is a foreign key item.

**Item type**

A decision was made to make item type its own entity in the hope that it will prevent repetition. The relationship between item and type is one to many because many items can be of one type but one item can be only of one type. Item TYPE is the parent and item is the child. The relationship is also non-identifying because item Type doesn’t need to be a primary key in item entity. The item type entity is also in a relationship one to many relationship with review, this will let the reviewer only review games.

**item & Supplier**

The Item and supplier entity relationship will need to be a many to many because the brief states that "The shop buys games, consoles and accessories from several suppliers". The supplier item bridge table will have supplieritemID as a primary key because supplier ID and item ID will depend on the supplier item ID. The decision was made to create this table to ensure that the shop will know where they can get each item.

**Item & Order**

The relationship between item and order is a many to many because the one item can be on many orders and many items can be on many items. Furthermore, creating a bridge table will help prevents repeating items from occurring on the table. A quantity attribute was added the item order bridge table. The decision was made to add the quantity attribute to this table because the quantity depend on the order number and the item.

**Order**

The order primary key is made of two attribute which makes it a composite key. This was necessary to meet the requirements the order number must start with a letter as its first character and then be followed by 5 numerical digits. The constraint (order\_alpha BETWEEN 'A' AND 'Z') was put on orderAlpha to ensure only letter can be inserted. The Sales rep and Order relationship is one to many because a rep can make many order but an order can only be made by one rep. This means that repID is a foreign key attribute in order entity.

**Delivery**

The decision was made to make delivery its own entity because a delivery doesn’t depend on what the order number. A delivery is separate. The relationship between order any delivery is many to many because many deliveries can be made for one order and one delivery can be made for many order. A delivery driver first name is added as an attribute but it has been left as NULL meaning it isn’t compulsory.

**Order & Delivery**

The order number and delivery id will be composite primary keys because the attribute delivery address depends on the two attributes. A customer number, contact name and address are added so that the delivery driver can find the delivery location and call customer if cannot safely deliver items.

**Reviewer**

Reviewer entity has the three attributes username, password and email. The username is set as the primary key because the reviewer needs to be uniquely identified when the log in the make a review. The email attribute has a validation constraint which is set up ensures all users insert an @ symbol. The email also is set up so it is UNIQUE meaning no user can enter the same email. To ensure that password is 8 length the precision of varchar2 is set to 8 characters long.

**Review**

Review entity has four attributes which include review ID (PK), item ID (FK), review date and review text(AK) which is set as unique to prevent reviewers plagiarising. It is important to have item type ID as a foreign key, this will display what game is being reviewed. The relationship between item type and review entities is one to many because many reviews can be made on one game. A constraint (itemTypeID = GAME) will be put on item type so that only game will be displayed.

**Reviewer & Review (ReviewSYSTEM bridge table)**

Reviewer and review entities have the relationship of many to many. A game can be reviewed by many reviewers and many games can be reviewed by many reviewers. The ReviewSYSTEM bridge table will stop anomalies from occuring. In bridge table the field number of reviews has been added with the constraint (NoOfReviews < 2). This constraint will ensure only one review is made on each game by each reviewer. The review ID and reviewer username will act as composite key because number of review depends on both fields to see if the game has been reviewed.