Music Shop

Jared Hayward (Team Coordinator)

Steve Miesch

Tyler Stout

Brendan Murray

CS 353-(01) Winter 2013

Tables of Contents

Pages 1-3 : Database Requirement Specifications

Page 4: ER Diagram

Page 5: Relational Schema and Integrity Constraint Table

Pages A1 – A22: project.out – printout file

**Database Requirements:**

**Nature of the Application:**

The goal of this database is to successfully store information and manage relationships relating to a chain of record stores. A record store, is a store that sells primarily music related products to customers such as CD’s, instruments, and other items such as posters or apparel related to music. There are multiple stores that are kept track of, each with their own set of employees. In addition the database will need to keep track of each store’s orders. An order contains a list of products being sold to a particular customer by an employee. These orders are managed by employees when a customer checks out, and the database will need to keep track of orders such as how many of a particular product is being sold at a particular store. One of the primary products being sold at these record shops are CD’s. Because of their popularity, CD’s will contain both the standard information related to all products, as well as their own unique set of information which is kept separate from the other products. Overall, the store will primarily keep track of stores, employees, orders, and products. In addition it will also keep track of CD’s as a subset of products.

**Workers:**

The database will contain information about employees related to their name, age, job title, and most importantly, their employee number which is used for keeping track of them in the database. In addition employees must work at a store, and an employee works at one store only. Employees are responsible for creating orders. An employee may handle multiple orders, however not every employee is required to process orders.

**Store:**

A store’s information includes its location, hours, and its store number which is used as a “key” to keep track of its information. Every store sells products but isn’t required to sell every product at their store, and a store may sell multiple products. In addition a store must have multiple employees working for it. A store also processes numerous orders, but a store isn’t required to process numerous orders.

**Order:**

The database tracks an order’s number which acts as the “key” as well as an order’s subtotal and the type of order. The type in this case is whether an order involves buying a product, returning a product, or reserving a product. Every order must be created by one employee, and the date on which the order is created must also be stored. Orders can only be processed through one store and they must be processed by a store. Orders must contain products, and they may contain more than one kind of product.

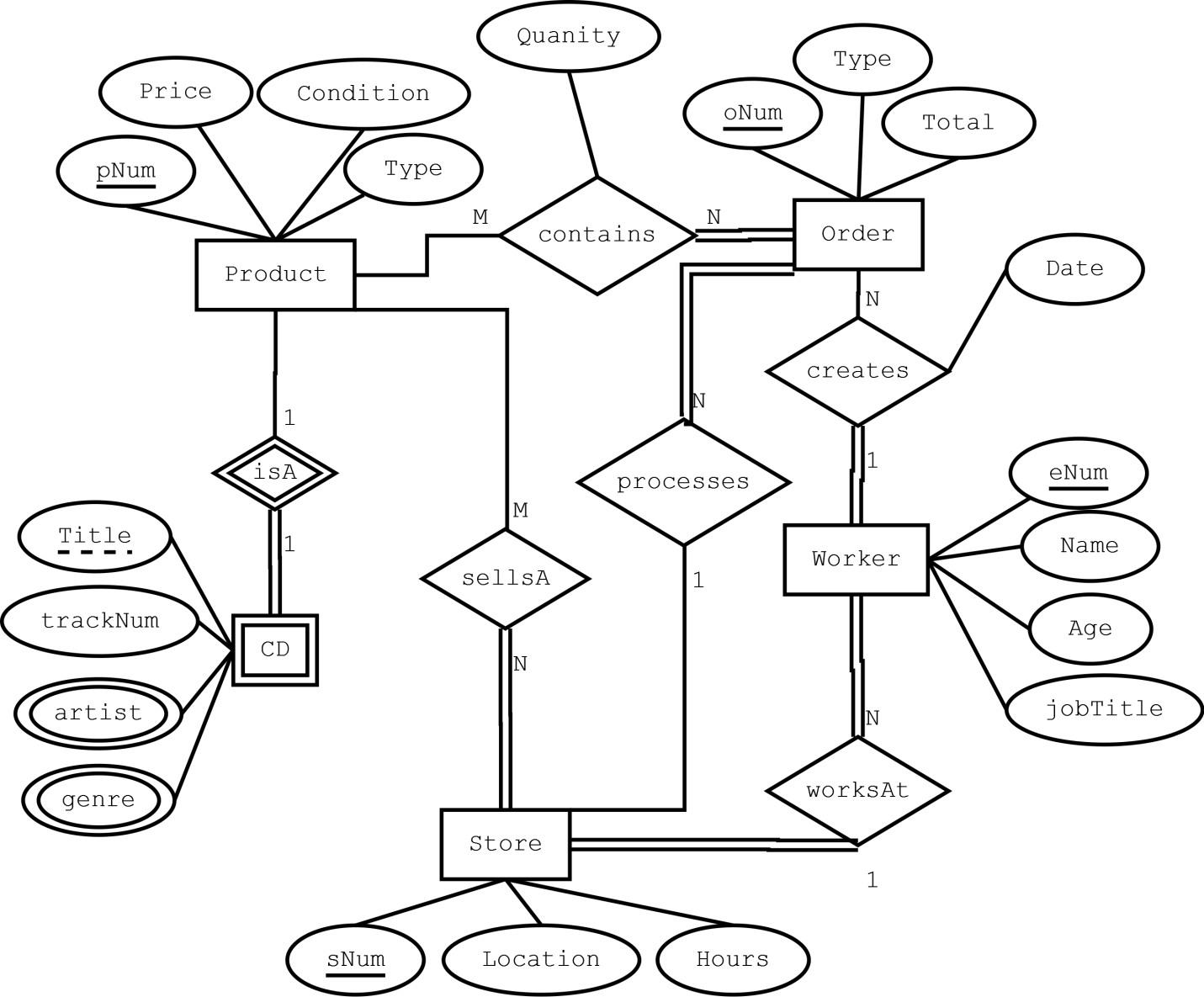
**Product:**

Information related to products includes the product’s unique number, a product’s price, its type, as well as its condition such as whether it is new or used. A product may be a CD. In addition multiple products may be sold, and they must be sold through stores. A product can potentially be found in multiple orders and in addition the quantity of a particular product in a particular order is also recorded.

**CD:**

A CD is another form of a product. Information that is unique to CD’s include its title which is partially responsible for keeping track of it, in addition to the number of tracks or songs found in each CD. In addition the database will also keep track of a CD’s artists in which a single CD may have multiple artists related to it. Information about the genre of music found on each CD is also kept track of, and one CD could potentially contain multiple genres of music. A CD must also be one kind of product. A store may have multiple CDs.

**ER Diagram:**



**Rudimentary Relational Schema:**

Product(pNum, Price, Condition, Type)

Orders(orderNum, eNum, oDate, sNum, Type, Total)

Store(sNum, Location, Hours)

Workers(eNum, sNum, Name, Age, jobTitle)

CD(Title, pNum, trackNum)

CD\_genre(genre, Title)

CD\_artist(artist, Title)

SellsA(sNum, pNum)

Contains(pNum, oNum, Quanity)

This relational schema is already in BCNF.

**Integrity Constraints:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IC Name & Table(s) | IC Type | English Statement | Page # where  implemented | Page #  where  tested |
| pIC1 – Table: Products | Key | This integrity constraint enforces that it is unique and not null. | Pg A1 | Pg A19 |
| wIC2 – Table: Workers | Foreign Key | This IC will use sNum as a foreign key from the Store table. To make sure each worker belongs to an existing store. | Pg A1 | Pg A19 |
| pIC2 | 1-attribute | Products Condition must be used new or refurbished. | Pg A1 | Pg A19 |
| wIC3 | 2-attribute, 1-row | An employee under 18 years of age can’t have a job title of manager. | Pg A1-A2 | Pg A20 |