**Name:**

**30 points**

**Objectives**

* Create an E-R diagram of the JRJ application database using the MySQL workbench reverse engineering tool
* Reverse engineering refers to taking an existing database and creating an E-R diagram
* Modify the E-R diagram for new application requirements
* Forward engineering is creating SQL DDL (create table statements) from an E-R diagram.
* Create and execute SQL create table statements for a new design

**Reading**

* Read chapter 10 in murach textbook on design, normalization and use MySQL Workbench for design.

**Database Setup**

* If you do not have the database schmea JRJ from prior weeks, download the file jrj\_create\_table.sql. This contains an sql script for creating schmea JRJ, tables and loading the tables with small amount of data.

**What to submit for this lab?**

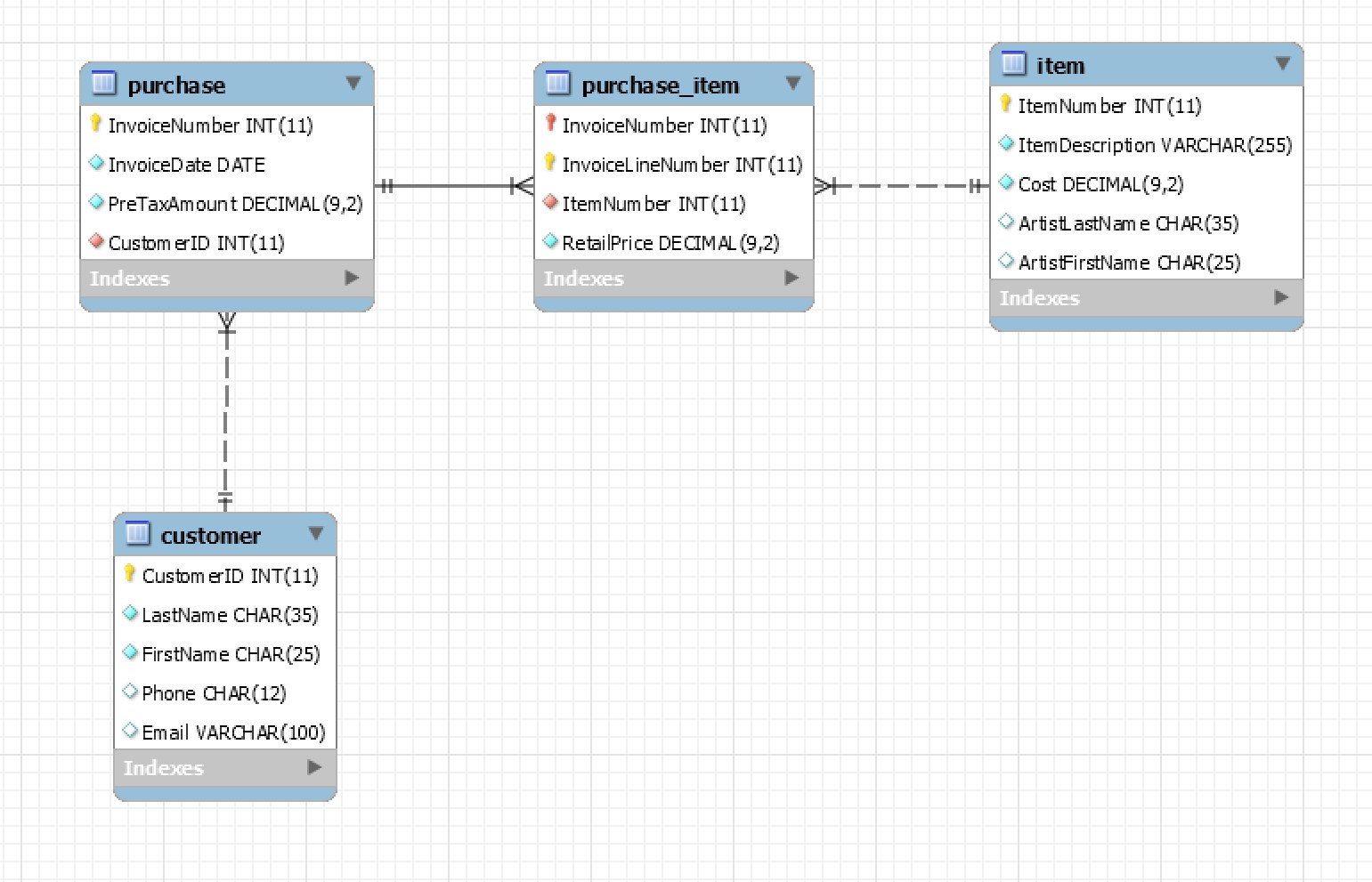
**Edit this lab document and save as PDF using Assign5\_<your last name>.pdf as the file name. Then submit to iLearn assignment 5.**

**Part 1. Reverse Engineering of the JRJ database**

Open MySQL Workbench connect to database server

Choose the menu option **Database 🡪 Reverse Engineer** and follow the prompts to select the tables in the JRJ schema.

When complete you should have an E-R diagram that looks like this



JRJ stands for James River Jewelry. It is a fictitious jewelry store. Items for sale are listed in the Item table. Profile information for customers is in the Customer table. When customers make a purchase, a row in the Purchase describes the overall purchase (the date of transaction, total amount). The items purchased are recorded in the Purchase\_Item table and include the price paid for this item.

Based on your reading of the E-R diagram, answer the following questions:

1. What is the key of the Customer entity?

-CustomerID is the key of the Customer entity

1. What is the key of the Purchase entity?

-InvoiceNumber is the key of the Purchase entity

1. Describe the relationship between Customer and Purchase?
   1. Is it 1-to-1, 1-to-Many, Many-to-Many?
      1. The relationship is 1-to-many. Each customer can make many purchases, but each purchase can only be made by one customer
   2. Is the relationship mandatory?
      1. Yes the relationship is mandatory
   3. Is a customer entity required to have at least one purchase?
      1. A customer entity is required to have a related purchase entity
   4. Is a purchase entity required to have a related customer entity?
      1. A purchase entity is required to have a related customer entity

1. What is the key of purchase\_item entity?
   1. InvoiceLineNumber

1. Describe the relationship between Purchase and purchase\_item?
   1. It is a 1-to-Many relationship. Each purchase can have one or more items purchased.

1. What is the key of Item entity?
   1. ItemNumber

1. Describe the relationship between purchase\_item and Item entities?
   1. It is a Many-to-1 relationship

**Part 2: Modify JRJ diagram for new requirements**

The Jewelry store wants to expand its database application to include

* tracking of artists and styles
* which customers are interested in artists and styles.
* The store wants to establish an award program.
* When a customer purchases 10 pieces of jewelry, the 11th purchase will receive a discount equal to 50% of the average of the 10 prior purchases.
* The store also wants to track how often and which pieces of jewelry are purchased using this award program.

There will new entities for Style and Artist. Customers can register their interest in particular styles and artists and so there are relationships between Customer entity and Style entity and between Customer and Artist entities.

Modify the E-R diagram for these new entities and relationships.

To implement the awards the customer entity will need attributes to track the number of purchases and the total of these purchases. When the number of purchases reaches 10, the next purchase will receive a discount equal to 50% of the average purchase with is 0.50\*(total/10). And the number of purchases and total will be reset to 0.

Modify the E-R diagram for these new attributes for the Customer entity.

Modify the Purchase\_Item entity to include a new field for AwardDiscount which is the discount (if any) given for this purchase.

1. Is the modified database design normalized? If yes, how do you know. If no, explain why you chose an unnormalized design.

1. Give an example of a design for Style and Artist data that is NOT normalized. What are some of the problems related to using designs that are not normalized?

1. If you were the database designer for this project, how would go about validating this new data model?

1. Save your design model for later.

**Part 3: Generate and review the SQL create statements**

* Menu 🡪 File 🡪 Export 🡪 Forward Engineer
* This will create an sql script. Save this script to a file.
* Review the sql script.

1. What is the definition of the column Purchase\_Item.AwardDiscount? Should the definition include the keywords NOT NULL? What do the keywords NOT NULL mean?

1. Can a customer register an interest in an artist if that artist does NOT exist in the Artist table? Explain why or why not?

1. If customers 1, 4 and 9 are interested in the artist Juan Pablo. These same customers are also interested in another artist Theresa Christi. How are these relationships represented in the database?

1. JRJ jewelry has a certain item for sale, but the artist is unknown. Do you see any problem when inserting the row for this item into the Item table?