**Referential Corruption**

Robert Mulligan | Jeff Zhao | Blake Kohn | Philip Lapid | Alexandra Niebrugge



S307: Spring, 2016

04/17/2016

**Table of Contents**

1 Executive Summary 2

2 Introduction 2

3 Design of the Database 3

3.1 Conceptual Design 3

3.2 Logical Design 8

3.3 Physical Design 9

4 Implementation of the Database 9

4.1 Create Tables SQL 9

4.2 Index SQL Code 15

5.1 Design 15

5.2 Implementation 16

**5.2.1 Reports** 16

**5.2.2 Forms** 17

6 User Manual 18

7 Conclusion 18

# 1 Executive Summary

Our database was built for IU Outdoor Adventures who is looking to automate and streamline their processes involving store sales, vendors, equipment, classes, trips, and employee payroll. After analyzing its operations, we’ve built a database to satisfy all necessary company functions, allowing IU Outdoor Adventures to build a technological interface that links all aspects of company operations.

We advised that IU Outdoor Adventures implement Oracle SQL Developer as their primary application to run and navigate their systems. The database was modeled based on the Enterprise Resource Diagram we developed for IU Outdoor Adventures. Our team built an Enterprise Resource diagram that represents the individual entities influencing or influenced by IU Outdoor Adventures’ operations. From the ER diagram, we built SQL statements and created dummy data to show the company what the database would look like with all of their current data fully populated in the database.

We believe that IU Outdoor Adventures will be able to utilize our assessment of its operations, Oracle system implementation, and maintenance procedures to further develop as a company and provide quality services for its customers.

# 2 Introduction

In order to fully satisfy the technological necessities of IU Outdoor Adventures, our team, Referential Corruption (name not indicative of project success or work ethic), has analyzed the operations within the company and provided them with a solution. We’ve included the database life cycle, possible opportunities, and solutions we foresee for IU Outdoor Adventures’ business needs.

Our team advised the company to implement Oracle SQL Developer (4.1.1). This system is a free integrated development environment that allows IU Outdoor Adventures to simplify both development and management of this type of database. The Developer offers end-to-end development, query application, and a migration platform.

The database life cycle has been included below to show the stages of the IU Outdoor Adventures’ assignment.

Database Life Cycle:

1. Planning - In the planning stage, we analyzed the operations of IU Outdoor Adventures and requested deliverables to further understand the functions involved.
2. Analysis - In the analysis phase, we recognized the gaps between the current system used and what was required of the systems. We did a requirements analysis and analyzed structuring opportunities through integrated conceptual modeling.
3. Logical Design - In the logical design stage, we decided the information requirements elicitation and structure and designed specifications of the database to be implemented.
4. Physical Design - In the physical design stage, we laid out technology necessities specific to IU Outdoor Adventures. This stage involved the physical database design and its programs.
5. Implementation - In the implementation stage, we implemented our written SQL for creating the database based off of the logical and physical designed specified earlier. From there, we ran the SQL code to ensure that it would work and deliver the correct information when needed. Next, we implemented the SQL code to create the database for IU Outdoor Adventures.
6. Maintenance - In the maintenance stage, we monitored our database to make sure that everything was performing the way that it was intended to. Following that, we completed a performance analysis to make sure that everything was performing at the highest level for the database and corrected any errors that were encountered. Lastly, we completed an audit of the database system to ensure accuracy of the database.

Our team built an ER diagram to display the entities involved in the company’s operations. We wrote SQL code to build tables displaying these entities and created “dummy data” to display what a fully populated database would look like for IU Outdoor Adventures. Many components were considered when analyzing and creating a solution for the company. These components are detailed in further sections of this report.

# 3 Design of the Database

## 3.1 Conceptual Design

Figure 1 - ER Diagram

Our ERD began with a standard student enrollment design for classes. Each course has many sections that have a variety of different dates available. This plethora of date options combined with the need to document the transactions between customers, employees, and IUOA gave rise to many quirks in our design. We determined that documentation was the most crucial part of IUOA’s operations since so many items are rented, bought, and sold on a daily basis. This led us to create multiple log tables to easily keep track of the specified requirements such as permits, equipment, transportation, and facilities.

Furthermore, the shop offers distinctly different product offerings ranging from merchandise and rentals to vacations and trips. To do this, we created a supertype in the products table to delineate the unique fields associated with each product category.

Lastly, we incorporated Point of Sale (POS) ledger capability to keep track of where sales occurred and then linked this table back to the employee table so show which employees sold which items. This would allow the ability to keep track of how much each employee sold as well as the hours that they logged to assist in the payroll process.

--- Field --     |    -- Type -- |        -- Nullable? -- |      -- Size -- |  -- Description --

----------------------------------------------------------------------------------------------------------------------------

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CUSTOMER\_ID | NUMBER | N | 22 | ID of Customer |
| CARDNUMBER | NUMBER | N | 22 | Number on card |
| CARDTYPE | VARCHAR2 | N | 20 | Credit or debit |
| EXP\_DATE | DATE | N | 7 | When card expires |

**Table 1 - Card Account**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TRAINING\_ID | NUMBER | N | 22 | ID of trainee |
| CERTIFICATION\_NAME | VARCHAR2 | Y | 20 | Name of certification earned |
| REQUIREMENTS | VARCHAR2 | Y | 20 | Requirements to meet this cert |

**Table 2 – Certifications**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COURSEID | VARCHAR2 | N | 10 | ID of the Course |
| COURSENAME | VARCHAR2 | N | 20 | Course Name |
| UNITS | VARCHAR2 | N | 20 | Credit hours |
| PRICE | VARCHAR2 | N | 20 | Cost of class |

**Table 3 – Courses**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PHONENUMBER | VARCHAR2 | Y | 20 | Phone number of customer |
| CUSTOMERNAME | VARCHAR2 | N | 20 | Name of customer |
| CUSTOMER\_ID | NUMBER | N | 22 | Unique Identifier |
| ADDRESS | VARCHAR2 | Y | 200 | Address of customer |
| CUSTOMERSINCE | DATE | Y | 7 | Date customer first made purchase |
| EMAIL | VARCHAR2 | Y | 100 | Email of customer |

**Table 4 – Customers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EMPLOYEE\_ID | NUMBER | N | 22 | Unique Identifier |
| EMPLOYEENAME | VARCHAR2 | N | 20 | Name of employee |
| BIRTHDATE | DATE | N | 7 | Date of birth |
| FULL\_TIME | VARCHAR2 | Y | 1 | Yes or No |
| WAGERATE | NUMBER | N | 22 | Hourly rate of employee |

**Table 5 – Employee**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EMPLOYEE\_ID | NUMBER | N | 22 | Unique Identifier |
| TRAINING\_ID | NUMBER | N | 22 | Unique Identifier |
| DATECOMPLETED | DATE | Y | 7 | When certificate was complete |

**Table 6 - Employee Training**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STUDENT\_ID | NUMBER | N | 22 | Unique Identifier |
| COURSEID | VARCHAR2 | N | 25 | Unique Identifier of the course |
| SEMESTER | VARCHAR2 | Y | 20 | What semester the course was in |
| ATTENDANCE | VARCHAR2 | Y | 20 | Day arrived in class |
| GRADE | VARCHAR2 | Y | 2 | What grade the student received |

**Table 7 – Enrollment**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TRIPID | VARCHAR2 | N | 20 | Unique identifier |
| EQUIPMENTID | NUMBER | N | 22 | Unique identifier |

**Table 8 - Equipment Log**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| COLOR | VARCHAR2 | Y | 20 | Color of equipment |
| EQUIPMENTNAME | VARCHAR2 | N | 50 | Name of equipment |
| EQUIPMENTID | NUMBER | N | 22 | Unique identifier |
| BRAND | VARCHAR2 | Y | 20 | Name of equipment producer |

**Table 9– Equipment**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FACILITY\_ID | NUMBER | N | 22 | Unique identifier |
| TRIPID | VARCHAR2 | N | 20 | Unique identifier |

**Table 10 - Facilities Log**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FACILITYNAME | VARCHAR2 | N | 50 | Name of facility |
| FACILITYLOCATION | VARCHAR2 | N | 20 | Location of facility |
| FACILITY\_ID | NUMBER | N | 22 | Unique Identifier |

**Table 11 – Facilities**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EMPLOYEE\_ID | NUMBER | N | 22 | Unique identifier |
| WORKDATE | DATE | N | 7 | Date of work |
| STATIONID | NUMBER | N | 22 | Unique identifier |
| HOURSWORKED | NUMBER | N | 22 | # of hours worked |
| PAYTYPE | VARCHAR2 | N | 6 | Method of payment |

**Table 12 - Hour Log**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SECTION\_ID | VARCHAR2 | N | 20 | Unique identifier |
| INSTRUCTORNAME | VARCHAR2 | N | 25 | Name of instructor |
| EMPLOYEE\_ID | NUMBER | N | 22 | Unique Identifier |

**Table 13 – Instructor**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MERCHANDISENAME | VARCHAR2 | N | 50 | Name of merchandise |
| MERCHANDISEID | NUMBER | N | 22 | Unique Identifier |

**Table 14 - Merchandise**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TRIPID | VARCHAR2 | N | 20 | Unique Identifier |
| PERMIT\_ID | NUMBER | N | 22 | Unique Identifier |

**Table 15 - Permits Log**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PERMIT\_ID | NUMBER | N | 22 | Permit ID |
| PERMITDURATION | DATE | N | 7 | Duration of permit |
| PERMITCOST | NUMBER | N | 22 | Cost of permit |
| PARKNAME | VARCHAR2 | N | 50 | Name of park |

**Table 16 – Permits**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADDRESS | VARCHAR2 | N | 50 | Station address |
| STATION\_LOCATION | VARCHAR2 | N | 40 | Location of station |
| STATIONID | NUMBER | N | 22 | Station ID # |
| PHONENUMBER | NUMBER | N | 22 | Phone Number |

**Table 17 - Point of Sale**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PRODUCT\_ID | NUMBER | N | 22 | Product’s unique identifier |
| VENDORID | NUMBER | N | 22 | Vendor # |
| WHOLESALECOST | NUMBER | N | 22 | Wholesale cost |
| CATALOGPRICE | NUMBER | N | 22 | Price of product |
| DESCRIPTION | VARCHAR2 | N | 100 | Description of product |
| PRODUCTTYPE | VARCHAR2 | N | 50 | The type of product |

**Table 18– Products**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RENTAL\_ID | NUMBER | N | 22 | Unique Identifier |
| RENTALNAME | VARCHAR2 | N | 20 | Name of who rented |
| RENTALDATE | DATE | N | 7 | Date of rental |
| DATERETURNED | DATE | Y | 7 | Date equipment returned |
| RETURNED | VARCHAR2 | N | 1 | Has equipment been returned |

**Table 19– Rentals**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SECTION\_ID | NUMBER | N | 22 | Unique Identifier |
| COURSEID | VARCHAR2 | N | 20 | Unique Identifier |
| ROOM\_NUMBER | VARCHAR2 | Y | 20 | Course Room |
| TRIPID | VARCHAR2 | N | 20 | Unique Identifier |

**Table 20– Sections**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STUDENT\_ID | NUMBER | N | 22 | Unique Identifier |
| STUDENTNAME | VARCHAR2 | N | 20 | Name of student |
| BIRTHDATE | DATE | N | 7 | Date of birth |
| MAJOR | VARCHAR2 | N | 20 | Major/Degree |
| ADDRESS | VARCHAR2 | N | 100 | Address |

**Table 21- Students**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TRANSACTIONID | NUMBER | N | 22 | Unique Identifier |
| TRANSACTIONDATE | DATE | N | 7 | Date of transaction |
| CUSTOMER\_ID | NUMBER | N | 22 | Unique Identifier |
| STATIONID | NUMBER | N | 22 | Unique Identifier |
| PRODUCT\_ID | NUMBER | N | 22 | Unique Identifier |

**Table 22– Transactions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TRIPID | VARCHAR2 | N | 20 | Unique Identifier |
| CAR\_ID | NUMBER | N | 22 | Unique Identifier |

**Table 23- Transportation Log**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CAR\_ID | NUMBER | N | 22 | Unique Identifier |
| CARTYPE | VARCHAR2 | N | 20 | Type of car |
| LICENSENUMBER | VARCHAR2 | N | 10 | License plate number |
| COLOR | VARCHAR2 | Y | 10 | Color of car |
| MAKE | VARCHAR2 | N | 20 | Brand of car |
| CAR\_MODEL | VARCHAR2 | N | 20 | Model of Car |
| CAR\_YEAR | NUMBER | N | 22 | Year of car model |

**Table 24– Transportation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TRIPID | VARCHAR2 | N | 10 | Unique Identifier |
| TRIPDATE | DATE | N | 7 | Date of trip |
| DESTINATION | VARCHAR2 | N | 40 | Destination |

**Table 25– Trips**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TRIPLOCATION | VARCHAR2 | N | 40 | Location of destination |
| TRIPDATE | DATE | N | 7 | Date of trip |
| VACATION\_ID | NUMBER | N | 22 | Unique Identifier |

**Table 26– Vacations**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VENDOR\_CATEGORY | VARCHAR2 | N | 25 | Type of goods provided |
| VENDORNAME | VARCHAR2 | N | 25 | Name of Vendor |
| CATEGORYDESCRIPTION | VARCHAR2 | Y | 150 | Description |
| VENDORID | NUMBER | N | 22 | Unique identifier of Vendor |

**Table 27 – Vendors**

## 3.2 Logical Design

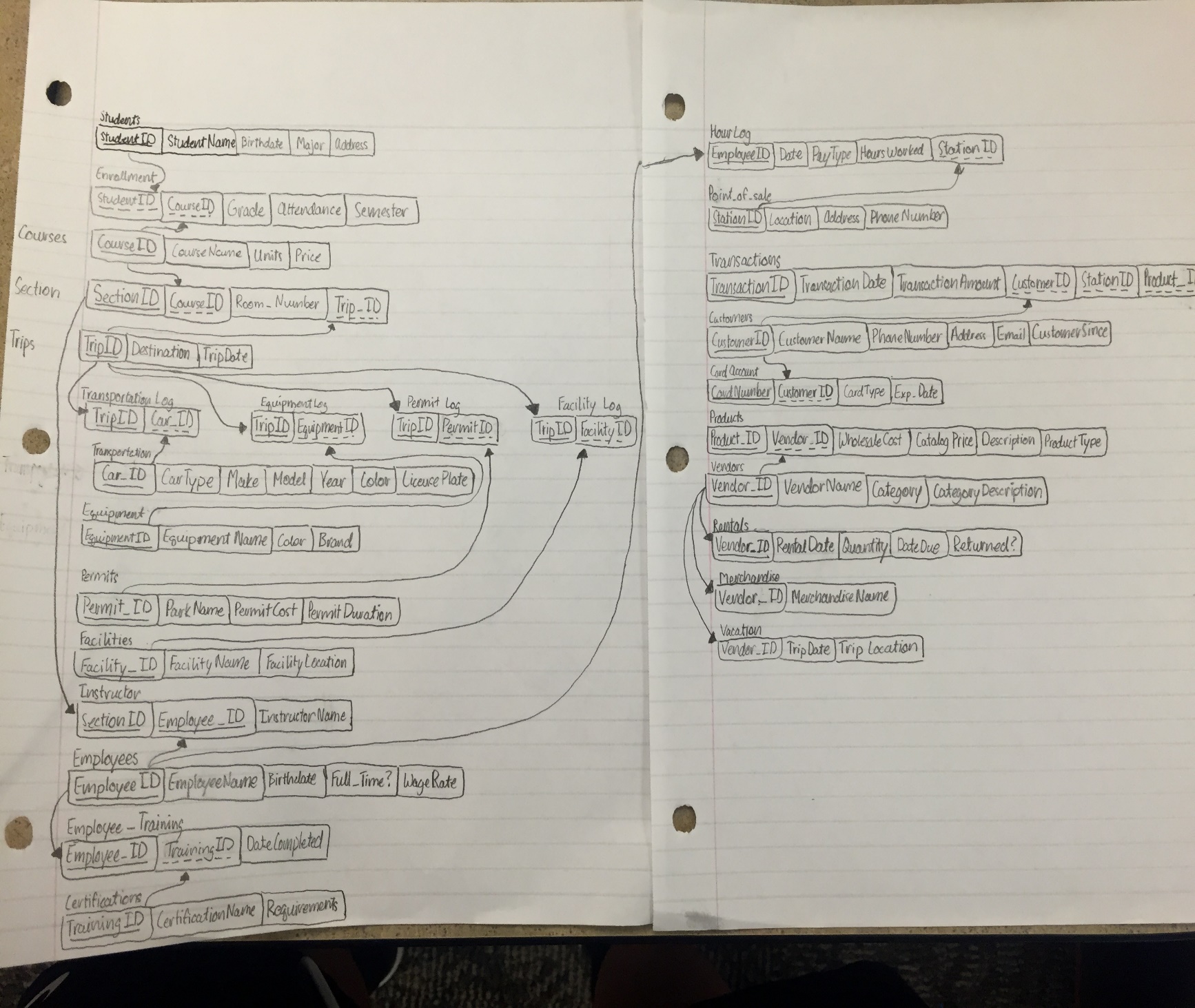


Figure 2 - Relational Schema

For IU Outdoor Adventures our team was able to create a relational schema that kept everything in third normal form. The only complication within the schema is related to the Transportation and Equipment tables. The only way to reach either of these tables through select statements are to be calling specifically from the tables themselves or going through several other tables to reach information within the two tables. An example of this would be a statement asking for information from a piece of Equipment used in a certain course Section. This means that the Section tables needs to go to the Trip table then the Transportation Log before it can reach information on equipment. We kept these tables in the positions that they are in because we were focusing on the transaction side of IUOA. Therefore the tables involving transactions are easily available to access information.

## 3.3 Physical Design

For our indexes, we chose to create a composite index on both the CUSOMTERS\_T and PRODUCTS\_T table. This organization revolves around customers, and there can be thousands of them. Staff may need to pull either ID or Name of a customer quickly as a lookup, thus requiring an index. IUOA has high volume of products and may need a quick price check for either retail sale or looking when negotiating purchase price with vendors.

# 4 Implementation of the Database

## 4.1 Create Tables SQL

/\*select 'drop table ' ||TABLE\_NAME|| ' cascade constraints;' from user\_tables;\*/

drop table EMPLOYEE\_T cascade constraints;

drop table STUDENTS\_T cascade constraints;

drop table POS\_T cascade constraints;

drop table HOURLOG\_T cascade constraints;

drop table COURSES\_T cascade constraints;

drop table TRANSPORTATION\_T cascade constraints;

drop table PERMITS\_T cascade constraints;

drop table FACILITIES\_T cascade constraints;

drop table VENDOR\_T cascade constraints;

drop table PRODUCTS\_T cascade constraints;

drop table RENTALS\_T cascade constraints;

drop table MERCHANDISE\_T cascade constraints;

drop table VACATION\_T cascade constraints;

drop table CERTIFICATIONS\_T cascade constraints;

drop table EQUIPMENT\_T cascade constraints;

drop table EMPLOYEE\_TRAINING\_T cascade constraints;

drop table CUSTOMERS\_T cascade constraints;

drop table ENROLLMENT\_T cascade constraints;

drop table TRIPS\_T cascade constraints;

drop table SECTIONS\_T cascade constraints;

drop table INSTRUCTOR\_T cascade constraints;

drop table CARD\_T cascade constraints;

drop table TRANSACTIONS\_T cascade constraints;

drop table TRANSPORTATION\_LOG\_T cascade constraints;

drop table EQUIPMENT\_LOG\_T cascade constraints;

drop table PERMITS\_LOG\_T CASCADE CONSTRAINTS;

DROP TABLE FACILITIES\_LOG\_T CASCADE CONSTRAINTS;

CREATE TABLE CUSTOMERS\_T

(Customer\_ID     NUMBER(10)     NOT NULL,

CustomerName    VARCHAR2(20)     NOT NULL,

PhoneNumber     varchar2(20),

Address         VARCHAR2(200),

Email            VARCHAR2(100),

CustomerSince     DATE,

CONSTRAINT CUSTOMERS\_PK PRIMARY KEY (Customer\_ID)

);

CREATE TABLE CARD\_T

(CardNumber     NUMBER(20)     NOT NULL,

Customer\_ID     NUMBER(10)     NOT NULL,

CardType  varchar(20) not null,

Exp\_Date         DATE         NOT NULL,

CONSTRAINT Card\_PK PRIMARY KEY (CardNumber),

CONSTRAINT CARD\_FK FOREIGN KEY (Customer\_ID) REFERENCES Customers\_T(Customer\_ID)

);

CREATE TABLE EMPLOYEE\_T

(Employee\_ID NUMBER(11,0) NOT NULL,

EmployeeName VARCHAR(20) NOT NULL,

Birthdate     DATE          NOT NULL,

Full\_time     varchar(1),

WageRate      NUMBER(6,2)   NOT NULL,

CONSTRAINT EMPLOYEE\_PK PRIMARY KEY (Employee\_ID)

);

CREATE TABLE STUDENTS\_T

(Student\_ID     NUMBER(10)     NOT NULL,

StudentName     VARCHAR(20)     NOT NULL,

Birthdate     DATE     NOT NULL,

Major         VARCHAR(20)     NOT NULL,

Address         VARCHAR(100)     NOT NULL,

CONSTRAINT STUDENTS\_PK PRIMARY KEY (Student\_ID));

CREATE TABLE POS\_T

(StationID       NUMBER(10)     NOT NULL,

Station\_Location varchar(40)    NOT NULL,

Address Varchar(50) not null,

PhoneNumber      NUMBER(15)      NOT NULL,

CONSTRAINT POS\_T PRIMARY KEY (StationID)

);

CREATE TABLE HOURLOG\_T

(Employee\_ID     NUMBER(10)     NOT NULL,

WorkDate         DATE         NOT NULL,

PayType         VARCHAR(6)     NOT NULL,

HoursWorked     NUMBER(2)     NOT NULL,

StationID     NUMBER(10)     NOT NULL,

CONSTRAINT HOURLOG\_FK FOREIGN KEY (Employee\_ID) REFERENCES EMPLOYEE\_T(Employee\_ID),

CONSTRAINT HOURLOG\_FK1 FOREIGN KEY (StationID) REFERENCES POS\_T(StationID)

);

CREATE TABLE COURSES\_T

(CourseID          VARCHAR(10)     NOT NULL,

COURSENAME     VARCHAR(20)     NOT NULL,

UNITS    VARCHAR(20)     NOT NULL,

PRICE VARCHAR(20) NOT NULL,

CONSTRAINT COURSE\_PK PRIMARY KEY (CourseID)

);

CREATE TABLE TRANSPORTATION\_T

(Car\_ID         NUMBER(10)     NOT NULL,

CarType         VARCHAR(20)     NOT NULL,

Make         VARCHAR(20)     NOT NULL,

Car\_Model         VARCHAR(20)     NOT NULL,

Car\_year         NUMBER(4)     NOT NULL,

Color         VARCHAR(10),

LicenseNumber    VARCHAR(10)     NOT NULL,

CONSTRAINT TRANSPORTATION\_PK PRIMARY KEY (Car\_ID)

);

CREATE TABLE PERMITS\_T

(Permit\_ID     NUMBER(10)     NOT NULL,

ParkName     VARCHAR(50)     NOT NULL,

PermitCost     NUMBER(20)     NOT NULL,

PermitDuration     DATE         NOT NULL,

CONSTRAINT PERMITS\_PK PRIMARY KEY (Permit\_ID)

);

CREATE TABLE FACILITIES\_T

(Facility\_ID     NUMBER(10)     NOT NULL,

FacilityName     VARCHAR(50)     NOT NULL,

FacilityLocation    VARCHAR(20)     NOT NULL,

CONSTRAINT FACILITIES\_PK PRIMARY KEY (Facility\_ID));

CREATE TABLE VENDOR\_T

(VendorID           NUMBER(10)     NOT NULL,

VendorName           VARCHAR(25)     NOT NULL,

Vendor\_Category     VARCHAR(25)     NOT NULL,

CategoryDescription VARCHAR(150),

CONSTRAINT VENDOR\_PK PRIMARY KEY (VendorID)

);

CREATE TABLE PRODUCTS\_T

(Product\_ID        NUMBER(10)   NOT NULL,

VendorID          NUMBER(10)    NOT NULL,

WholesaleCost     NUMBER(8)     NOT NULL,

CatalogPrice      NUMBER(8)     NOT NULL,

Description       VARCHAR(100)  NOT NULL,

ProductType       VARCHAR(50)     NOT NULL,

CONSTRAINT PRODUCTS\_PK PRIMARY KEY (Product\_ID),

CONSTRAINT PRODUCTS\_FK FOREIGN KEY (VendorID) REFERENCES VENDOR\_T(VendorID)

);

CREATE TABLE RENTALS\_T

(Rental\_ID     NUMBER(10)    NOT NULL,

RentalName     VARCHAR(20)   NOT NULL,

RentalDate     DATE       NOT NULL,

DateReturned   DATE,

Returned VARCHAR(1) not null,

CONSTRAINT RENTALS\_PK FOREIGN KEY (Rental\_ID) REFERENCES PRODUCTS\_T(PRODUCT\_ID)

);

CREATE TABLE MERCHANDISE\_T

(MerchandiseID     NUMBER(10)   NOT NULL,

MerchandiseName varchar(50)         NOT NULL,

CONSTRAINT MERCHANDISE\_FK FOREIGN KEY (MerchandiseID) REFERENCES PRODUCTS\_T(Product\_ID)

);

CREATE TABLE VACATION\_T

(Vacation\_ID     NUMBER(10)     NOT NULL,

TripDate         DATE           NOT NULL,

TripLocation     varchar(40)    NOT NULL,

CONSTRAINT VACATION\_FK FOREIGN KEY (Vacation\_ID) REFERENCES PRODUCTS\_T(Product\_ID)

);

CREATE TABLE CERTIFICATIONS\_T

(Training\_ID      NUMBER(10)      NOT NULL,

Certification\_Name VARCHAR(20),

Requirements       VARCHAR(20),

CONSTRAINT CERTIFICATIONS\_PK Primary KEY (Training\_ID)

);

CREATE TABLE EQUIPMENT\_T

(EquipmentID     NUMBER(10)     NOT NULL,

EquipmentName     varchar(50)     NOT NULL,

Color varchar(20),

Brand varchar(20),

CONSTRAINT EQUIPMENT\_PK PRIMARY KEY (EquipmentID)

);

CREATE TABLE EMPLOYEE\_TRAINING\_T

(Employee\_ID     NUMBER(10)       NOT NULL,

Training\_ID       NUMBER(10)      NOT NULL,

DateCompleted     DATE,

CONSTRAINT EMPLOYEE\_TRAINING\_FK1 FOREIGN KEY (Employee\_ID) REFERENCES EMPLOYEE\_T(Employee\_ID),

CONSTRAINT EMPLOYEE\_TRAINING\_FK2 FOREIGN KEY (Training\_ID) REFERENCES CERTIFICATIONS\_T(Training\_ID));

CREATE TABLE TRANSACTIONS\_T

(TransactionID     NUMBER(10)     NOT NULL,

TransactionDate     DATE  NOT NULL,

Customer\_ID     NUMBER(10)     NOT NULL,

StationID     NUMBER(10)     NOT NULL,

Product\_ID     NUMBER(10)     NOT NULL,

CONSTRAINT TRANSACTIONS\_PK PRIMARY KEY (TransactionID),

CONSTRAINT TRANSACTIONS\_FK1 FOREIGN KEY (StationID) REFERENCES pos\_T(StationID),

CONSTRAINT TRANSACTIONS\_FK2 FOREIGN KEY (Product\_ID) REFERENCES PRODUCTS\_T(PRODUCT\_ID),

CONSTRAINT TRANSACTIONS\_FK3 FOREIGN KEY (customer\_ID) REFERENCES Customers\_T(customer\_ID)

);

CREATE TABLE ENROLLMENT\_T

(Student\_ID     NUMBER(10)     NOT NULL,

CourseID     VARCHAR(25)     NOT NULL,

Grade         VARCHAR(2),

Attendance     VARCHAR(20),

Semester  Varchar(20),

CONSTRAINT ENROLLMENT\_FK1 FOREIGN KEY (Student\_ID) REFERENCES STUDENTS\_T(Student\_ID),

CONSTRAINT ENROLLMENT\_FK2 FOREIGN KEY (CourseID) REFERENCES COURSES\_T(CourseID)

);

CREATE TABLE TRIPS\_T

(TripID     VARCHAR(10)         NOT NULL,

TripDate  DATE  NOT NULL,

Destination     VARCHAR(40)     NOT NULL,

CONSTRAINT TRIPS\_PK PRIMARY KEY (TripID)

);

CREATE TABLE SECTIONS\_T

(Section\_ID     NUMBER(10)     NOT NULL,

CourseID     VARCHAR(20)     NOT NULL,

Room\_Number  VARCHAR(20),

TripID  Varchar(20) not null,

CONSTRAINT SECTIONS\_PK PRIMARY KEY (Section\_ID),

CONSTRAINT SECTIONS\_FK1 FOREIGN KEY (CourseID) REFERENCES COURSES\_T(CourseID),

CONSTRAINT SECTIONS\_FK2 FOREIGN KEY (TripID) REFERENCES TRIPS\_T(TripID)

);

CREATE TABLE INSTRUCTOR\_T

(Section\_ID     VARCHAR(20)     NOT NULL,

Employee\_ID     NUMBER(20)     NOT NULL,

InstructorName     VARCHAR(25)     NOT NULL,

CONSTRAINT INSTRUCTOR\_PK PRIMARY KEY (Section\_ID),

CONSTRAINT INSTRUCTOR\_FK FOREIGN KEY (Employee\_ID) REFERENCES EMPLOYEE\_T(Employee\_ID)

);

CREATE TABLE TRANSPORTATION\_LOG\_T

(TripID  Varchar(20) not null,

Car\_ID         NUMBER(10)     NOT NULL,

CONSTRAINT TRANSPORTATION\_LOG\_FK1 FOREIGN KEY (TRIPID) REFERENCES TRIPS\_T(TRIPID),

CONSTRAINT TRANSPORTATION\_LOG\_FK2 FOREIGN KEY (CAR\_ID) REFERENCES TRANSPORTATION\_T(CAR\_ID)

);

CREATE TABLE EQUIPMENT\_LOG\_T

(TripID  Varchar(20) not null,

EquipmentID     NUMBER(10)     NOT NULL,

CONSTRAINT EQUIPMENT\_LOG\_FK1 FOREIGN KEY (TRIPID) REFERENCES TRIPS\_T(TRIPID),

CONSTRAINT EQUIPMENT\_LOG\_FK2 FOREIGN KEY (EQUIPMENTID) REFERENCES EQUIPMENT\_T(EQUIPMENTID)

);

CREATE TABLE PERMITS\_LOG\_T

(TripID  Varchar(20) not null,

Permit\_ID     NUMBER(10)     NOT NULL,

CONSTRAINT PERMITS\_LOG\_FK1 FOREIGN KEY (TRIPID) REFERENCES TRIPS\_T(TRIPID),

CONSTRAINT PERMITS\_LOG\_FK2 FOREIGN KEY (PERMIT\_ID) REFERENCES PERMITS\_T(PERMIT\_ID)

);

CREATE TABLE FACILITIES\_LOG\_T

(TripID  Varchar(20) not null,

Facility\_ID     NUMBER(10)     NOT NULL,

CONSTRAINT FACILITIES\_LOG\_FK1 FOREIGN KEY (TRIPID) REFERENCES TRIPS\_T(TRIPID),

CONSTRAINT FACILITIES\_LOG\_FK2 FOREIGN KEY (FACILITY\_ID) REFERENCES FACILITIES\_T(FACILITY\_ID)

);

## 4.2 Index SQL Code

DROP INDEX CUSTOMERS\_T;

DROP INDEX PRODUCTS\_T;

CREATE INDEX CUSTOMERS\_INDEX ON CUSTOMERS\_T (Customer\_ID, CustomerName );

CREATE INDEX PRODUCTS\_INDEX ON PRODUCTS\_T (Product\_ID, Product\_Category, VendorID, WholesaleCost, CatalogPrice) ;

**5 User Interface**

## 5.1 Design

The targeted users of this database are the employees of IU Outdoor Adventure and IU auditors. The employees would be using the database for internal use and the IU auditors

What the targeted users may want to do with the database

* See how much equipment belongs to each vendor, as well as the amount IU Outdoor Adventure pays to each vendor
* See how much of each equipment they have, as well as how much of each equipment is currently being rented and used on a current trip
* See how many customers have attended a certain number of classes and trips

The front end interface will have different parts of the system. It will consist of categories related to IU Outdoor Adventure such as vendor, students, employees, classes, equipment, and more. From there, they will be able to see all the information for each category as well as go further into it to see by a certain type of equipment or the equipment that they bought from a specific vendor.

## 5.2 Implementation

### **5.2.1 Reports**

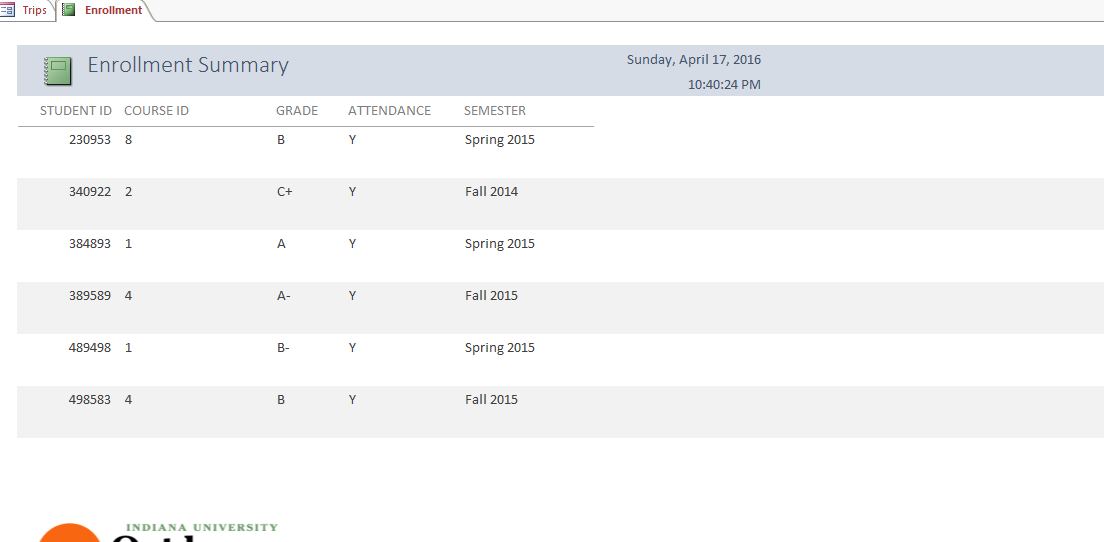


Figure 3 - Enrollment Report

This screenshot shows a report of the students enrolled in each course and semester. It gives details about attendance and grade earned for each student ID (safer than including student names especially when considering FERPA regulations).

Figure 4 - Employee Report

This is a screen shot of a report of IU Outdoor Adventure’s employees and their information for each employee such as their name, birthdate, if they are full-time or not, and what their hourly pay is.

### **5.2.2 Forms**

Figure 5 - Vendor Information Form

This is a screen shot of a form for vendors. It has the vendor information as well as the products, price, and the product information in the form. The form also has the information on rentals of products.

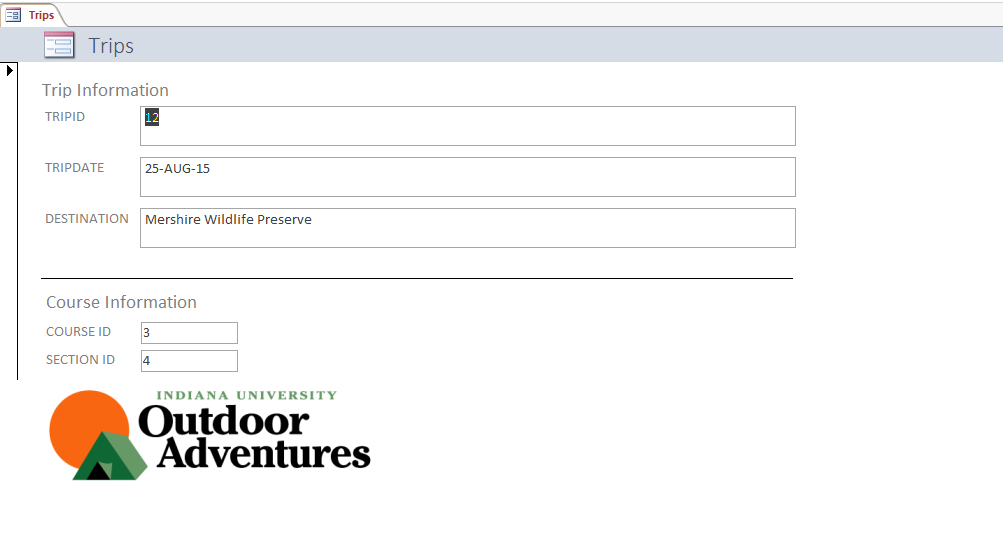


Figure 6 - Trip Form

This screenshot shows a form that displays trip information. It shows each trip and the corresponding class. Students can only attend trips for classes in which they are enrolled. This form shows details of the trip and the class sections.

# 6 User Manual

Include:

* Screen shots (use Alt+PrintScreen to capture a screen and Ctrl+V to paste); briefly describe the screen shots.
* Include “interesting” VB code that is related to database access only.

This database for IU Outdoor Adventures is going to be solely used internally by employees of IU Outdoor Adventures. The employees will use the database for a few reasons. They will use it for their equipment, vendor management, employee payroll, classes and trips, and store sales.

First, the employees will use the database to see the number of hours each instructor has worked. This will let them search by an instructor name to see a total amount of hours worked and more specifically, the number of hours worked for each class. This will give the employees a better idea who is working a lot and who is not working as much as they potentially should be. In addition, they can see which instructors have completed certain types of training and got their certifications. This will help IU Outdoor Adventures know which instructors are certified for certain classes and trips.

Second, the employees of IU Outdoor Adventures will use the database to know what kinds of equipment they have and how much of each equipment they have as well. Following that, they will be able to search what type of equipment is being used on each trip, so they will know where their inventory is being used, as well as how much inventory they have available to use.

Third, the employees of IU Outdoor Adventures will use the database to see how much the quantity of equipment from each vendor, as well as the costs for each equipment and the costs from each vendor. This will help the employees know which vendors they are ordering a lot of equipment from and which vendors they are spending the most money on.

Fourth, the employees of IU Outdoor Adventures will use the database to search for specific types of students based on certain criteria. For example, they could see which customers have attended at least four classes over the last year.

Next, the employees of IU Outdoor and Adventures will use the database for their equipment rentals. They will use this to know which equipment items are being rented, the date of rental, and when the equipment is expected back. Using this will let the employees have a better control of their inventory.

Overall, this database will be used solely by the employees of IU Outdoor Adventures. They will use it to have better control of their equipment through rentals, where each equipment is coming from in terms of their vendors, and how much of each equipment they have overall. With this database, they will also be able to know more about their customers and students which will let them be able to know specifics about each customer and student.

# 7 Conclusion

Our team was hired to deliver a solution for IU Outdoor Adventures. After strong analysis of the company and careful consideration of possible software options, we advised the company to implement Oracle SQL Developer to provide for their business needs. We built an ER diagram depicting their processes and entities and build the SQL to display data similar to that of the company.

While the project inherently was successful for IU Outdoor Adventures, our team ran into a few adversities when completing the project implementation. While our team did take the time to analyze the inner workings of the company, we ran into some issues later in time that could have been avoided had we spent a little more careful time analyzing the inner workings of the company. Our project timeline also was not as efficient as it could have been. We worked through a lot of the code in the first part but forgot to consider user interfaces early on when distinguishing components of the project. There was a level of scope creep involved as well because our team’s scope began too narrow from the outset but incorporated many more components as time went on. Overall, our team ran into few and infrequent issues when undergoing the project, many of which can be attributed to lack of expertise specific to our project.

Each of our team members have some type of technology based major, and this project allowed us to utilize our knowledge to provide a better solution for a local company, IU Outdoor Adventures. While we all have worked with numerous databases in the past, this was the first time that we were building one of our own, and we were able to see exactly what time and effort, opportunities and weaknesses are involved in database creation and user interfaces. Individually developing each aspect of the code, diagrams, etc. was beneficial to the project, but became all the more rewarding when we were able to link these components all together in deliverables such as forms and reports. Our team had a successful experience and enjoyed working with IU Outdoor Adventures to find a solution to satisfy their business needs.