This document presents an example of the deliverable due for the Database Design Project assignment. Most of the steps and sub-steps present an example of what you should provide in your project deliverable. Some of the steps and sub-steps (sub-step 1.4, for example) provide an explanation of what, if anything, you need to develop. You can easily distinguish an example from an explanation; explanations are written in *itlaics.*

Step 1: Create and check the E-R model

**1.1 Identify entities**

|  |  |
| --- | --- |
| **Entity** | **Description** |
| Location | Information regarding the business location |
| Car | Information regarding the rental car |
| Customer | Information regarding the customer |
| Employee | Information regarding the staff |
| Car Class | Information regarding the class of car |
| Rental Agreement | Information regarding the rental details |
| Discount | Information regarding the discounts/promotions applied to rental price |

**1.2 Identify relationships**

|  |  |  |
| --- | --- | --- |
| **Entity** | **Relationship** | **Entity** |
| Employee | Manages | Location |
| Location | Has | Employee |
| Location | Has | Car |
| Location | Services | Customer |
| Customer | Agrees to | Rental Agreement |
| Discount | Given to | Customer |
| Discount | Applied to | Rental Agreement |
| Car | Belongs to | Car Class |



**1.3 Identify and associate attributes with entities, identifying attribute domains and primary keys**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity** | **Attribute Name** | **Description** | **Data Type & Length** | **Nulls** | **Multi-valued** |
| Location | LocationID\*\* | Uniquely identifies each rental location | 4 fixed characters | No | No |
|  | Street | Street address of rental location | 60 variable characters | No | No |
|  | City | City of rental location | 25 variable characters | No | No |
|  | State | State of rental location | 2 fixed characters | No | No |
|  | Zip | Zipcode of rental location | 5 variable characters | No | No |
| Customer | DL\_Number\*\* | Uniquely identifies each customer with driver’s license number | 15 variable characters | No | No |
|  | First\_Name | First name of customer | 40 variable characters | No | No |
|  | Last\_Name | Last name of customer | 40 variable characters | No | No |
|  | Phone | Phone numbers of customer | 10 variable characters | No | Yes |
|  | Email | Email address of customer | 255 variable characters | No | Yes |
|  | Street | Street address of customer | 255 variable characters | No | No |
|  | City | City of customer | 40 variable characters | No | No |
|  | State | State of customer | 2 variable characters | No | No |
|  | tZip | Zipcode of customer | 5 variable characters | No | No |
| Employee | DL\_Number\*\* | Uniquely identifies each employee with driver’s license number | 15 variable characters | No | No |
|  | Location\_ID | Office location of employee | 4 fixed characters | No | No |
|  | Position | Job position of employee | 40 variable characters | No | No |
|  | First\_Name | First name of employee | 40 variable characters | No | No |
|  | Last\_Name | Last name of employee | 40 variable characters | No | No |
|  | Phone | Phone numbers of employee | 10 variable characters | No | Yes |
|  | Street | Street address of employee | 255 variable characters | No | No |
|  | City | City of employee | 40 variable characters | No | No |
|  | State | State of employee | 2 variable characters | No | No |
|  | Zip | Zipcode of employee | 5 variable characters | No | No |
| Car | Car\_ID\*\* | Uniquely identifies each car | 9 variable characters | No | No |
|  | Car\_Class\* | Car class | 5 fixed characters | No | No |
|  | Make | Car make | 40 variable characters | No | No |
|  | Model | Car model | 40 variable characters | No | No |
|  | Year | Car year | 4 fixed characters | No | No |
|  | Color | Car color | 40 variable characters | No | No |
|  | License\_Plate | Uniquely identifies each car by license plate | 8 variable characters | No | No |
|  | Current\_Location | Rental Location where car currently resides | 4 fixed characters | No | No |
| Rental Agreement | Rental\_Num\*\* | Uniquely identifies each rental agreement | 9 variable characters | No | No |
|  | DL\_Number\* | Uniquely identifies each customer with driver’s license number | 15 variable characters | No | No |
|  | Car\_ID\* | Uniquely identifies each car | 9 variable characters | No | No |
|  | Location\_Rented | Rental pick up location | 4 fixed characters | No | No |
|  | Location\_Returned | Rental drop off location | 4 fixed characters | No | No |
|  | Date\_Rented | Date of rental start | 8 fixed characters | No | No |
|  | Date\_Returned | Date of rental end | 8 fixed characters | No | No |
|  | Rental\_Status | Status of rental | 2 fixed characters | No | No |
|  | Fuel\_at\_Return | Fuel at rental end | 4 variable characters | No | No |
|  | Odometer\_Rented | Odometer reading at rental start | 6 fixed characters | No | No |
|  | Odometer\_Returned | Odometer reading at rental end | 6 fixed characters | No | No |
|  | Rental\_Rate | Calculated rental cost calculated from rental period, car class, and discounts charged to customer | 6 variable characters | No | No |
| Discount | Upgrade | Flat rate discount when a better value car is given | 6 variable characters | No | No |
|  | Promotion | Percent discount for running promotions | 3 variable characters | No | No |
|  | EmployeeDiscount | Percent discount for employees | 3 variable characters | No | No |
| Car Class | Car\_Class\*\* | Car class | 5 fixed characters | No | No |
|  | Class\_Price | Daily rental cost for car class | 6 variable characters | No | No |

1.4 Check that the model supports user transactions



Step 2 Map the E-R model to tables

2.1 Create tables

2.2 Check table structures using normalization

*Make modifications to the table from stem 2.1 to bring all tables in third normal form compliance.* ***Reproduce the entire table, including all tables, whether the table was modified or not.***

2.3 Check that the tables support user transactions

2.4 Check that integrity constraints are addressed

*Document that all integrity constraints are considered in the design. Be sure to address Domain, Entity, and Referential integrity constraints*

Step 3 Select database management system

*For this step, you do not actually implement a database. You do need to make a decision regarding the database management system(DBMS) you would select for implementation (i.e. Oracle, Microsoft SQL Server, Microsoft Access, MySQL, etc.). As a deliverable for this step, you will need to provide a discussion of the DBMS you would select. Your discussion must include:*

1. *Identify the DBMS selected*

We are recommending that the car rental company use Oracle for the DBMS. It is the number 1 DBMS out there and provides class leading support and due its popularity, finding people to support is easier and more cost effective to support upwards scalability and ongoing maintenance. The application can be made to run on a cloud environment, allowing for quick scale up/down and better uptime. As the company grows or additional tables are needed they will be best supported by Oracle.

1. *Identify the alternative DBMS(s) (you must consider at least two DBMSs) considered.*

The alternate DBMS considered that were Microsoft SQL and MySQL

1. *Discussion of the rational for the DBMS selected, including*
	1. *The strengths identified for the DBMS selected The biggest reason to choose oracle is the long term cost to maintain and update is far less due to the ease of use of the platform and the amount IT support to support it over time. Additionally, the greatest functionality is provided by the Oracle solution that reduces time spent making changes. .*
	2. *The strengths identified for the DBMS(s) not selected Both MYSQL and Microsoft SQL have a much lower cost of entry compared to Oracle and now that MYSQL is owned by Oracle support would be just as good.*
	3. *The weaknesses identified for the DBMS selected- The initial cost for Oracle is the major downfall for the DBMS. It is far greater than the other 2.*
2. *The weaknesses identified for the DBMS(s) not selected- For Microsoft SQL* does not scale up well at all. It will require multiple db servers to run our application efficiently. The IIS webserver frontend will also be a cow, requiring its own IIS farm .MySQL still has some maturity issues such as difficult data recovery from crashes, default settings that limit performance, and poor scaling. Some of these issues are taking a long time to get resolves ever since Oracle took over MySQL; many community developers have stopped working on the code.