**CMPE 226 Project Report**



**Submitted by - Team Datamass**

**Ashwini Balaraman (010139246)**

**Manasvini Banavara Suryanarayana (010102040)**

**Samhita Pyla (010734802)**

**Shubhra Gupta (010703797)**

**Submitted to:**

**Professor Ron Mak**

**Table of contents**

1. Application Introduction ………………….........................................................................2
2. Data Sources ……………………………………………………………………………...3

2.1 Operational Db Creation ……………………………………………………………...3

2.2 Analytical Db Creation ……………………………………………………………….4

3. Data Models ……………………………………………………………………………….6

3.1 Operational Db Data Model …………………………………………………………..6

3.2 Analytical Db Data Model ……………………………………………………………8

4. ER Diagram ………………………………………………………………………………..9

5. Relational Schema ………………………………………………………………………...11

6. Star Schema ………………………………………………………………………………12

7. Technology Stack Used …………………………………………………………………...12

8. Queries ……………………………………………………………………………………..2

8.1 Operational Queries ……………………………………………………………………2

8.2 Analytical Queries ……………………………………………………………………16

9. POC (Cis Server) …………………………………………………………………………..2

9.1 Configuration …………………………………………………………………………..2

9.2 Operations ……………………………………………………………………………...2

1. **Application Introduction**

We have developed a car rental reservation system application, which will be used by customers renting the car service and also by the employees of different car rental services. Our application provides two interfaces, one for customer and another for admin providing the rental car service.

The application is responsive and interactive with the users in a way that, each customer can find the car of their choice from various rental stores and regions and their by making the reservation of the car for a period of time. While booking the car, personal information of the user will be captured and stored in a database, which will be used by the rental service employees for various operations.

On the other hand, the admin interface of the application is used by employees of rental service for pulling the data of customers and their booking information. Search menu is provided in the application dashboard for admin users to perform extensive search on the data set present in the database. The side navigations provided in the application is used to pull different analytical and operational query results. Time based comparison of sales are available across regions, car types etc., which can be analyzed by admin to improve their car rental sales.

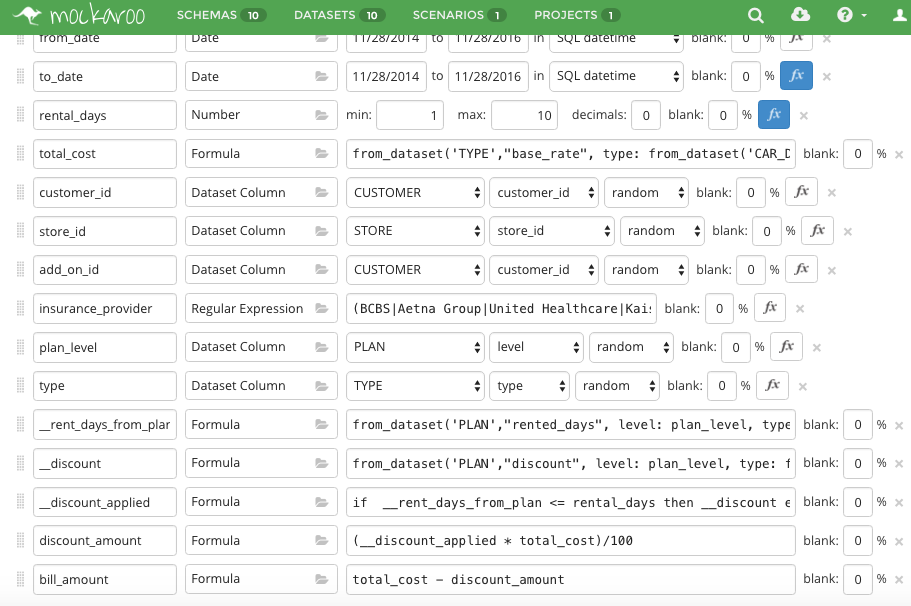
The data set of the application contains both analytical and operation data stored. The data set includes information regarding car models, brands, prices and information on employees, customers reserving the car, each reservation details, plans available with each transaction etc.

1. **Data Sources**

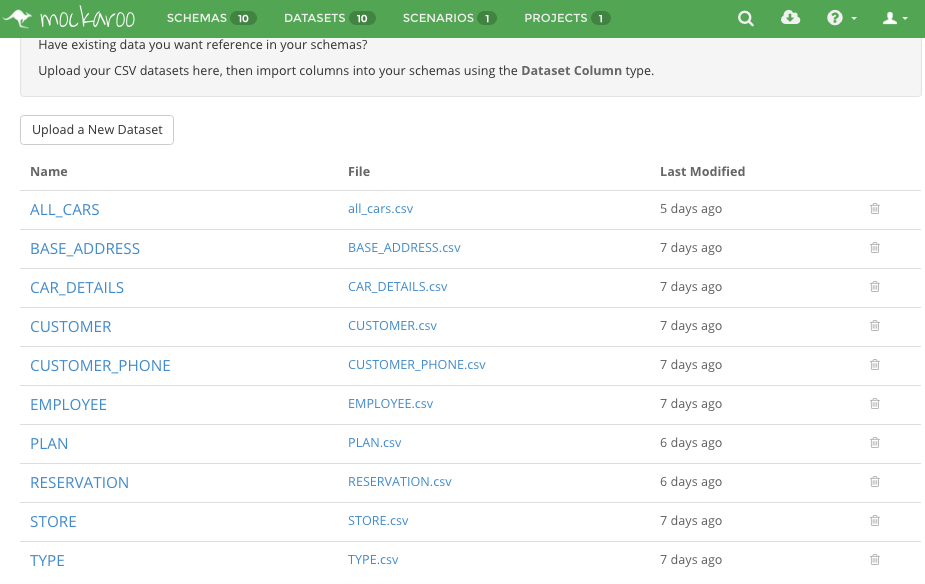
**2.1 Operational Db Creation**

We used Mockaroo which is an online data generation tool. It lets us create datasets and upload them. We can create other table data by referring to the uploaded datasets for foreign key references. It offers various types of data such as credit card numbers, insurance providers, states, cities, people names, phone numbers etc. Mockaroo also allows us to use formulae and choose their distributions.

**Data Creation**



**Uploading Datasets**



**2.2 Analytical Db Creation**

The analytical database is a specialized database design to support data analysis and business intelligence. It contains historic data of the operational transactions. The data stored in analytical database helps to predict trends of sales, profits, user-preferences over a longer period of time. The analytical database stores summarized data for transactions occurring on operational database. The data is stored once physically and hardly changes. End users can’t modify analytical tables. Only administrative users for the application are allowed to query the analytical database to understand the business progress and trends over many years. The query involves fetching and processing large amounts of rows and data. Due to its, less frequent queries and large amounts of data, analytical database tables are stored in different database than operational database.

For our application, analytical database needed to be in synchronization with the operational database. After creating operational database using online tool, we created analytical database using MySQL queries and stored procedures.

Some of the queries involved for creation of dimensional tables.

*INSERT INTO m\_analytical\_db.dim\_car (car\_vin, car\_id, brand, model, type, no\_of\_passengers, no\_of\_bags)*

*SELECT a.car\_vin, c.car\_id, c.brand, c.model, c.type, c.no\_of\_passengers, c.no\_of\_bags*

*FROM datamass.all\_cars a, datamass.car\_details c*

*WHERE a.car\_id = c.car\_id*

*INSERT INTO m\_analytical\_db.dim\_time (full\_date, day\_of\_week, day\_of\_month, month, quarter, year)*

*SELECT r.from\_date, 1, 1, 1, 'Q1', 2016*

*FROM datamass.reservation r*

*UPDATE dim\_time SET month = MONTH(full\_date)*

*UPDATE dim\_time SET day\_of\_month = DAYOFMONTH(full\_date)*

*DELIMITER //*

*CREATE PROCEDURE Quarter(*

*IN month int(2),*

*OUT quarter varchar(2) )*

*BEGIN*

*IF month < 4 THEN*

*SET quarter = 'Q1';*

*ELSEIF (month >= 4 AND month <= 6) THEN*

*SET quarter = 'Q2';*

*ELSEIF (month >= 7 AND month <= 9) THEN*

*SET quarter = 'Q3';*

*ELSEIF month > 9 THEN*

*SET quarter = 'Q4';*

*END IF;*

*SELECT quarter;*

*END//*

*DELIMITER ;*

*INSERT INTO m\_analytical\_db.fact\_reservation\_bookings (reservation\_id, bill\_amount, rental\_days, store\_key, car\_key, cust\_key, time\_key)*

*SELECT r.reservation\_id, r.bill\_amount, r.rental\_days, s.store\_key, c.car\_key, cs.cust\_key, t.time\_key*

*FROM datamass.reservation r, m\_analytical\_db.dim\_store s, m\_analytical\_db.dim\_car c, m\_analytical\_db.dim\_customer cs, m\_analytical\_db.dim\_time t*

*WHERE r.store\_id = s.store\_id AND r.car\_vin = c.car\_vin AND r.customer\_id = cs.cust\_id AND r.from\_date = t.full\_date*

**3. Data Models**

**3.1 Operational Db Data Model**

The operational database is also called as the Online Transactional Processing Database or OLTP , as it allows operations like create, insert, update and delete to be performed on data in real time. This type of databases provide high availability, fault tolerance and replication or scaling out.All these supported distributed databases.

Their major use is to maintain real-time business information.

Our application “car rental system” requires the following tables as part of our operation database for real-time operations:

**car\_details** - this table has all the details of the different cars available in the rental system.It has a *car\_id* as the *primary key* to identify each of the different car models present.Correspondingly the different car *brands*, their *model*, *type* and information like *no\_if\_passengers* and *no\_of\_bags* are in this table.Here, the type is a *foreign key* referenced from the *type table*.

**customer** - this table contains information related to the customer. It has the field like *customer\_id* which is the *primary key* and uniquely identifies each of the customer.It also has the user’s first\_name,last\_name,email,gender,points,license\_no,credit\_card\_no.

**employee** - this table has the *employee\_id* as *primary key*.It has *first\_name,last\_name,salary, designation and store\_id*.Here, store\_id is a *foreign key* referred from the store table.

**store** - this table has a *store\_id* as the *primary key*.It has the *street,site\_no, no\_of\_employees* and *address\_id* is a *foreign key* referenced from the *base\_address table.*

**base\_address** - this table has the *address\_id* as the *primary key* which is referenced in the other related tables.It has *city, state* and *zip* fields related to the location of the stores.

**all\_cars** - this table has the *car\_vin* as the primary key and it has the details of the cars at all the stores with the help of the car\_details and store\_id references. It also has availability and *car\_id,store\_id* are *foreign keys* referenced from the *car\_details table* and *store table* respectively.

**customer\_phone** - this table has the user *phone* number as the primary key with the corresponding *customer\_id*’s referred from the *customer\_details table.*

**reservation** - this table has a *reservation\_id* which is a *primary key* and is uniquely available for each reservation made.It also has *from\_date, to\_date, rental\_days,total\_cost,insurance\_provider,*

*Discount\_amount,total\_cost,car\_vin* referenced from the *all\_cars table,customer\_id* from *customer\_details table,store\_id* from the *store table,add\_on\_id* is an additional driver who is also a customer from the *customer\_details table* and another *foreign key* referenced from it, *plan\_level* from the *plan table* and *type* from the *type table.*

**plan** - this table has *plan and type*(both together) as a *composite primary key*.It also has discount and rented days.

**type** - this table has *type* as it’s *primary key* and base\_rate as it’s features.

**3.2 Analytical Db Data Model -** Our analytical database contains four dimensional tables and two fact tables that enable to analyze our sales for bookings and investment done on purchasing cars. The surrogate key for each dimensional table is an auto increment integer that increments for each row added.

The dimension tables for our analytical data model are as follows :-

i) dim\_car – This table is created from the all\_cars and car\_details tables from the operational tables. This dimension table contains all information about each car, including its vin number, brand, model, space, type and other details.

ii) dim\_customer – This dimensional table contains all information about the customers who have taken our service and is directly created from table customer in operational database.

iii) dim\_store – This table contains information about all the stores and locations used in the application. The data is dimensional table is populated from store and base\_address tables in operational database.

iv) dim\_time – This is the most important dimension as it will help up to identify and analyze operations over different periods of time. This table contains data from the reservation booking date from reservation table in operational database. The day\_of\_month, quarter and other attributes are filled using additional queries.

The application has 2 fact tables :-

i) fact\_reservation\_bookings – The fact table is created using surrogate keys from dim\_store, dim\_customer, dim\_car and dim\_time. It has attributes for the total bill amount and rental days for a reservation. This fact table enables to keep track of complete information of reservation bookings and analyze user preference for bookings.

ii) fact\_car\_purchase – This fact table contains surrogate keys from dim\_store, dim\_car, time\_key. It summarizes the cost of purchasing a car by any store. This information helps us to keep track of the amount spent on purchasing different cars.

**4. ER Diagram**

The entity-relationship diagram below represents the relationships between various entities of our car rental system. Entities are the objects containing the data. Cardinality between the entities shows the level of participation of each entity in the relationship. Following are the entities of our car rental reservation system.

* Address
* Employee
* Store
* Customer
* CustomerPhone
* Reservation
* Car
* All\_Cars
* Type
* Plan

Relationships between entities:

* Each Store has a location of type Address
* Each Store contains many Cars
* Each Reservation chooses Store
* Multiple Employees works in a Store
* Customer in each Reservation reserves a car
* Customer has phone number
* Each Car is valid for a type of offer Plan
* Each Reservation might be applied different Plans

|  |
| --- |
|  |

**5. Relational Schema**

The relational diagram shows the different entities and relationships in the database system without any of the anomalies and satisfying all the normal forms.It has named relations defined by a set of attributes. A relational schema has the following properties:

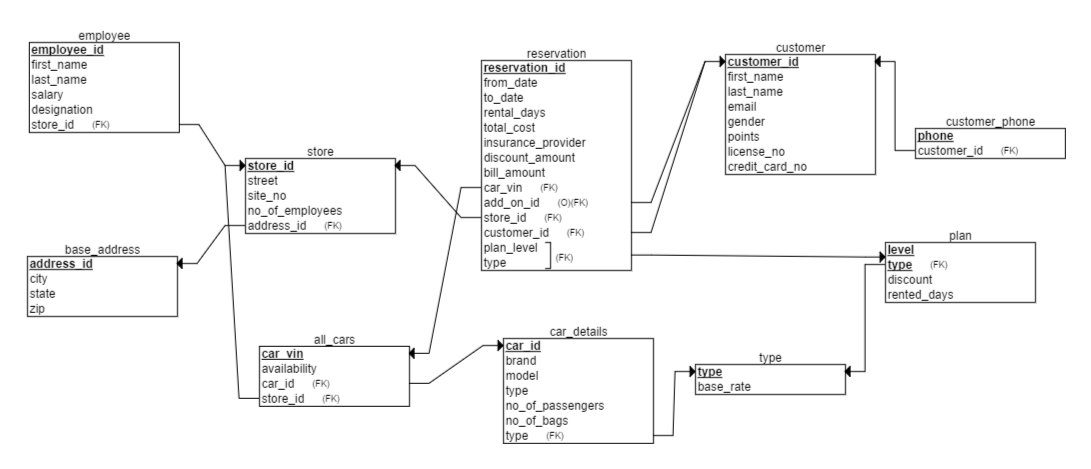
-entities should have distinct names

-every attribute in a relationship should have distinct names

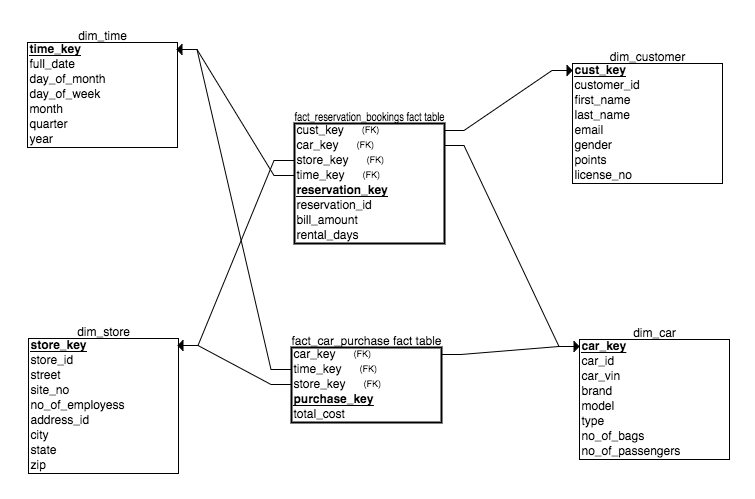
-there should not be duplicate tuples

-order of the attributes do not have significance

Our entities are represented in tables and their attributes are represented inside the tables with the primary key attributes underlined and other keys specified by a letter in the parenthesis ().



**6. Star Schema -** The below start schema shows the links between fact tables and dimension tables that represents our analytical database.



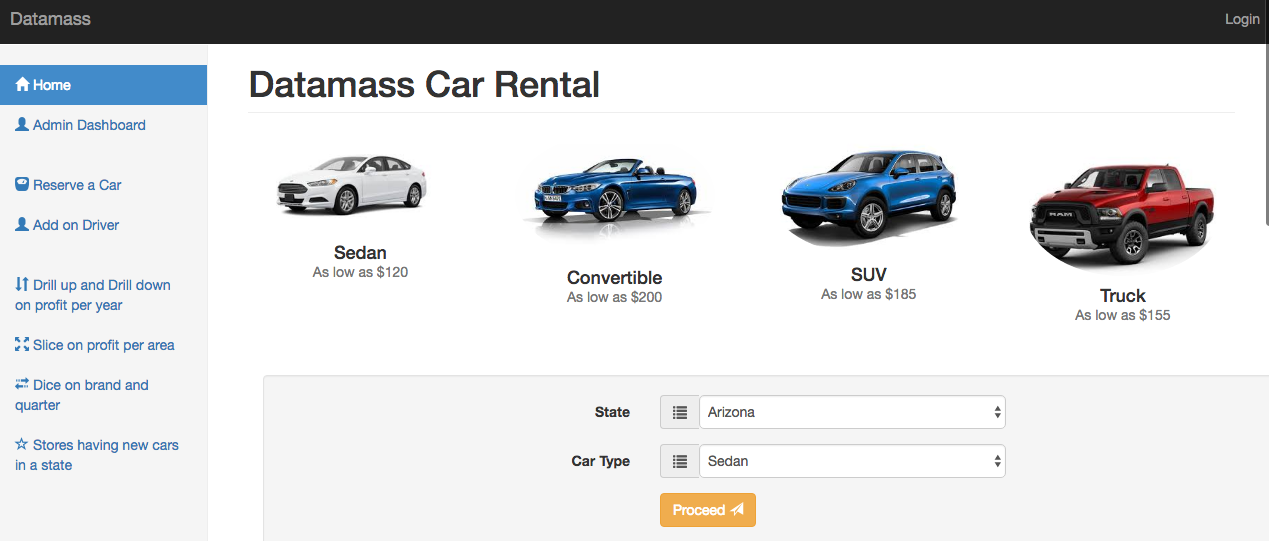
**7. Technology Stack Used**

* PHP
* Bootstrap
* HTML
* CSS
* AngularJS
* MySql queries
* Database set up on PhpMyAdmin
* Data set created using Mockaroo.com

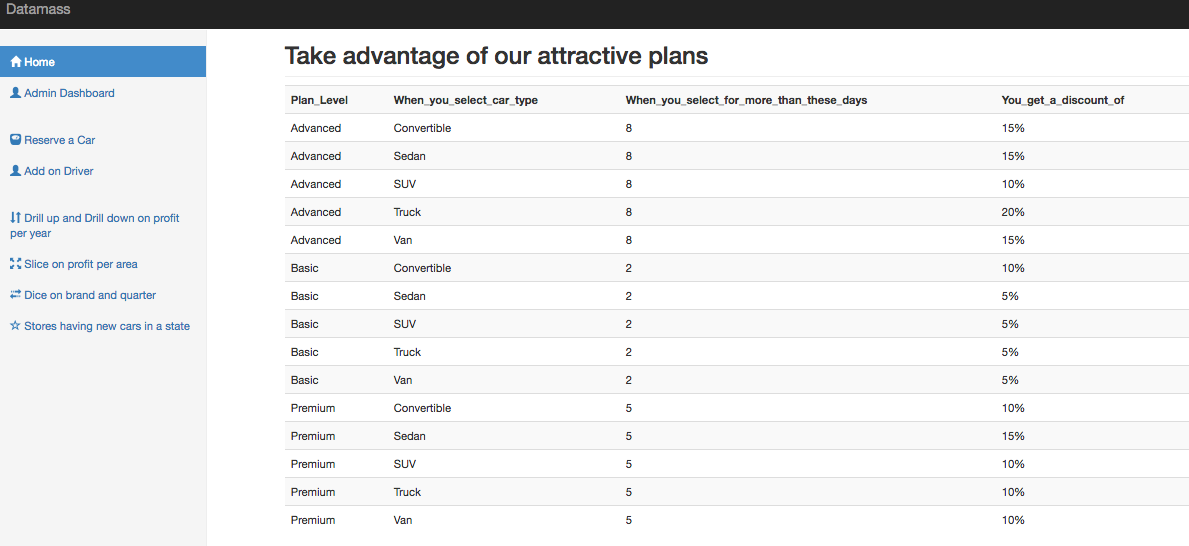
**8. Queries**

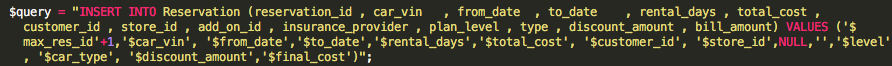
**8.1 Operational Queries**

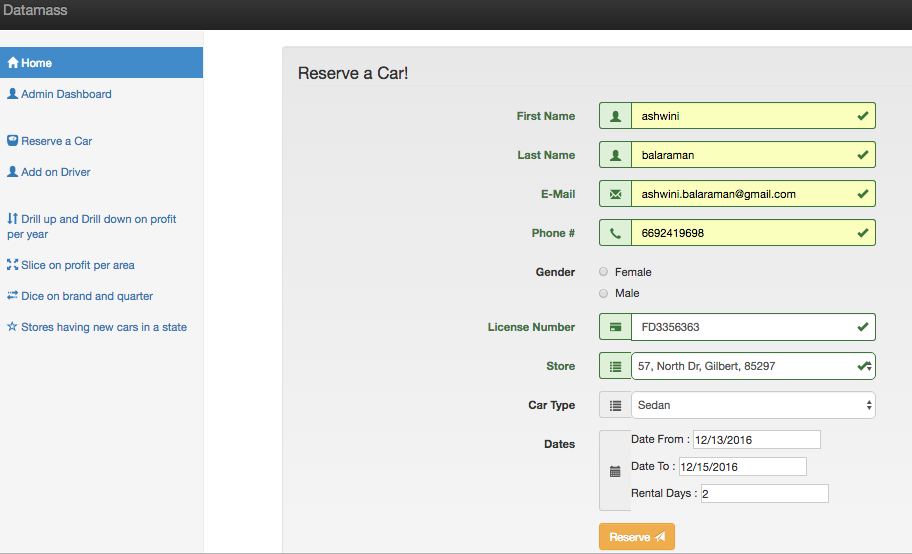


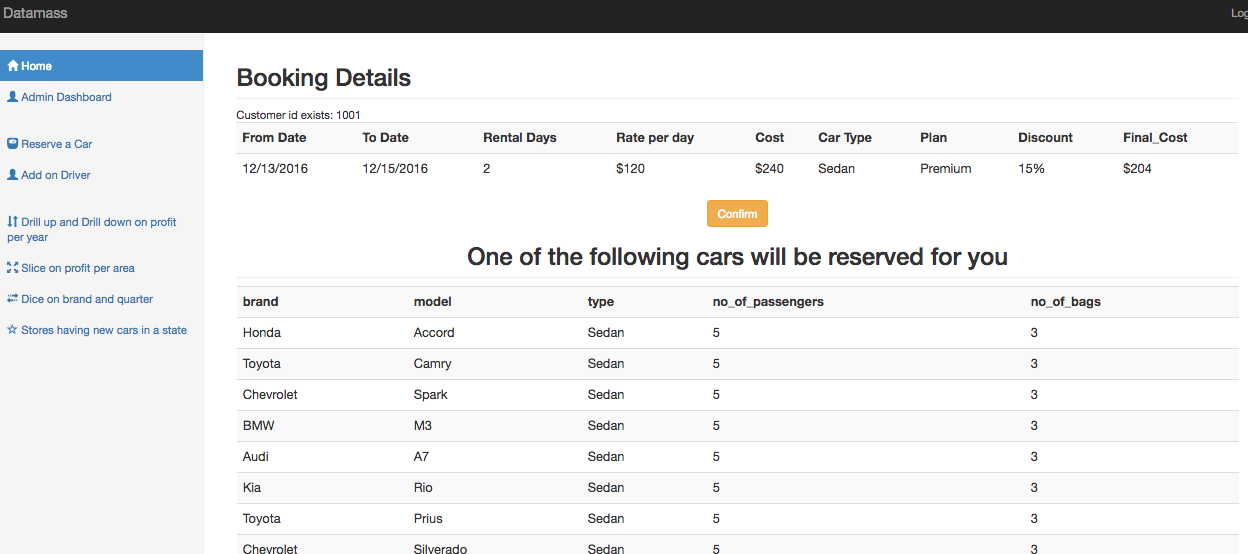


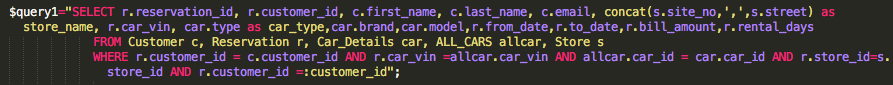


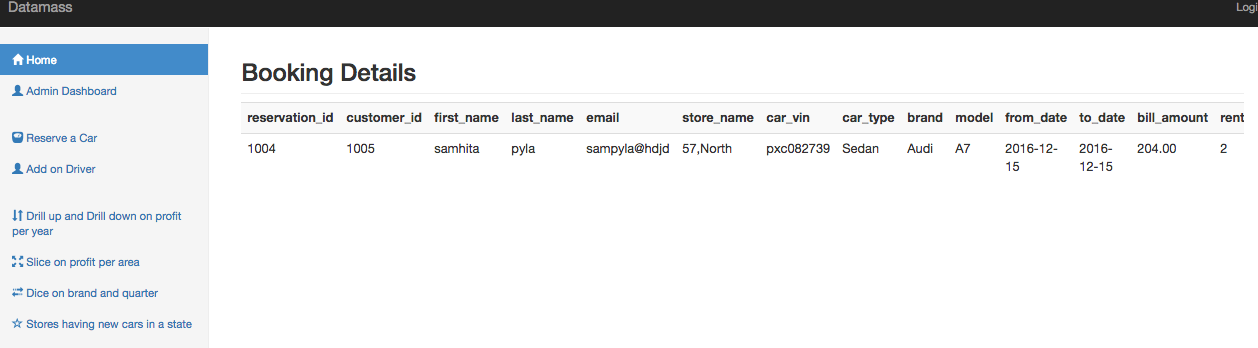




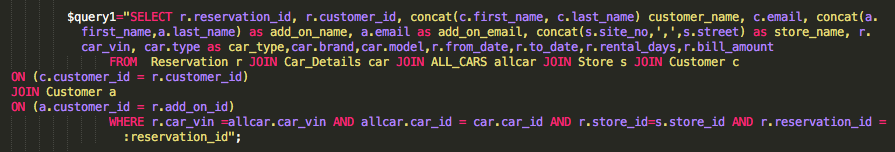


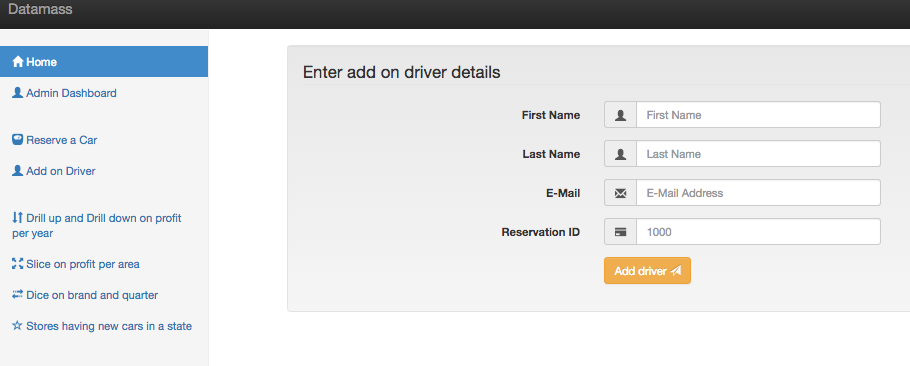








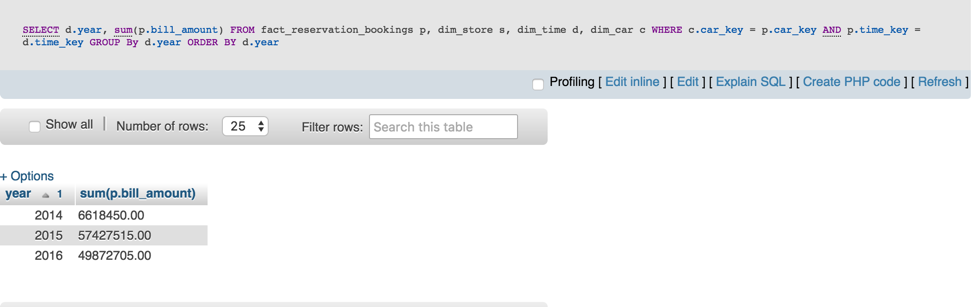




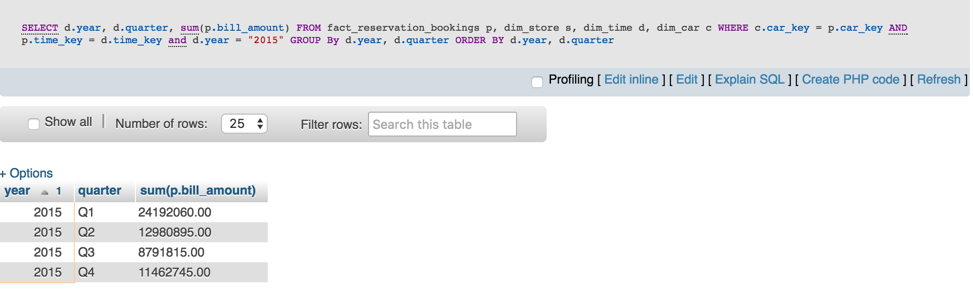
**8.2 Analytical Queries**

We extracted the information from the operational tables and transformed into the combined fields in analytical queries.

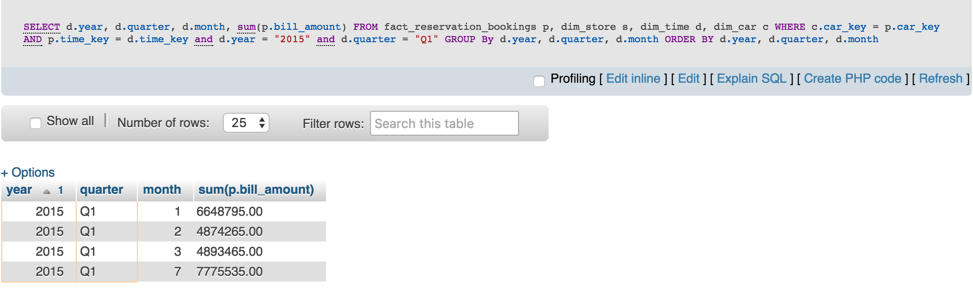
Drill down/Drill up operations:

Yearly profit query:

Drilling down on 2015:

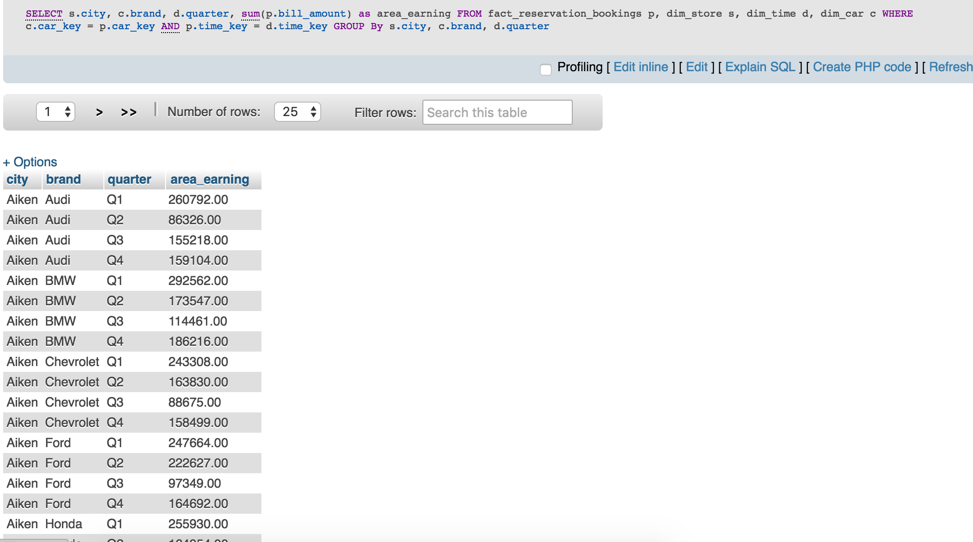


Drilling down on quarter Q1:

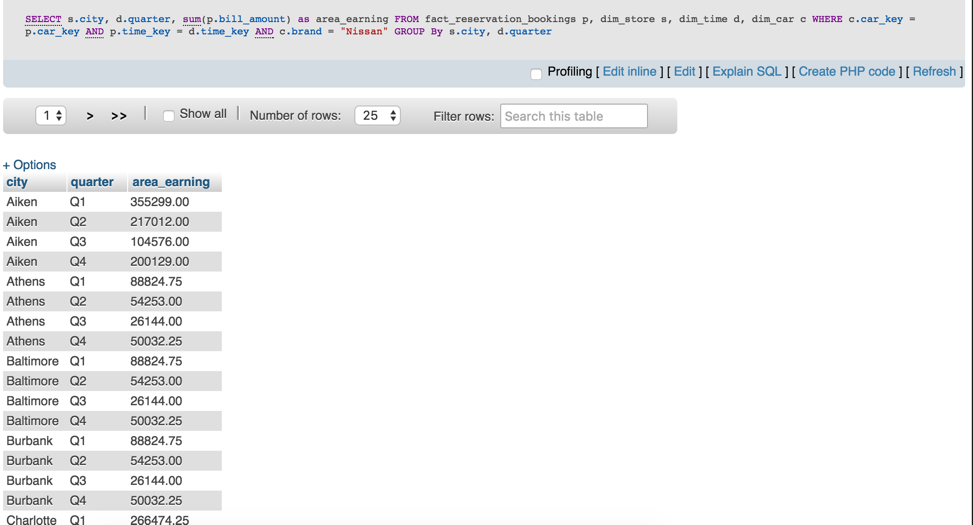


Slice/Dice operations:

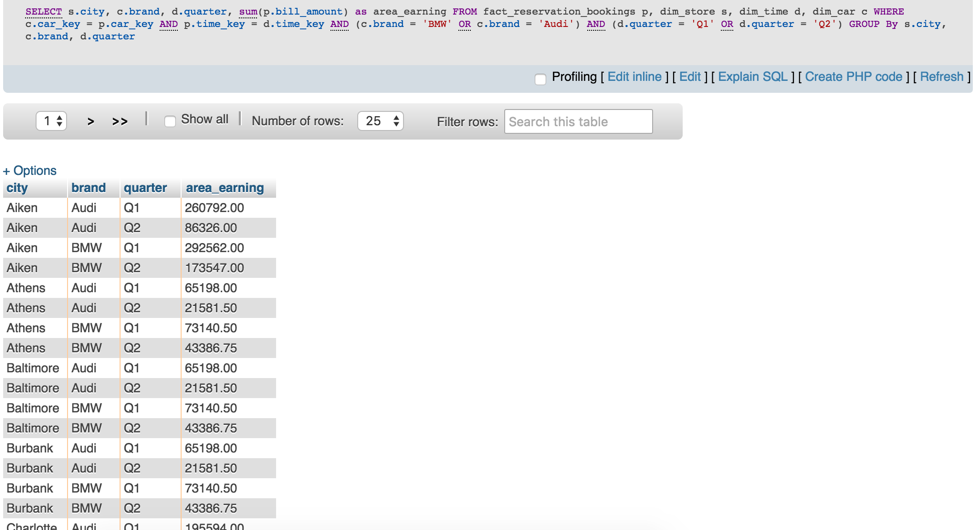
Profit per city and brand in quarters:



Slice on brand “Nissan”:



Dice on Q1, Q2 quarters and BMW, Audi brands:



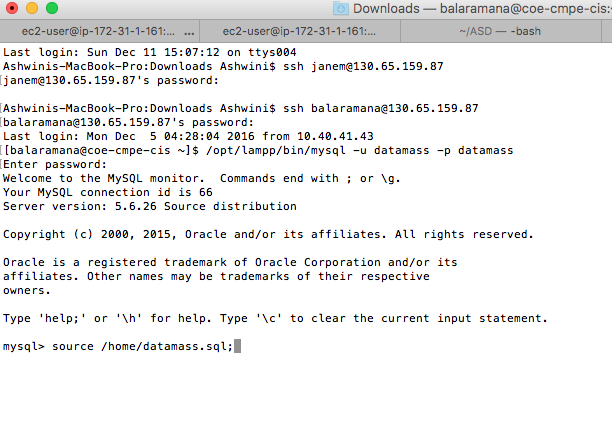
**9. POC (Cis Server)**

**9.1 Configuration**

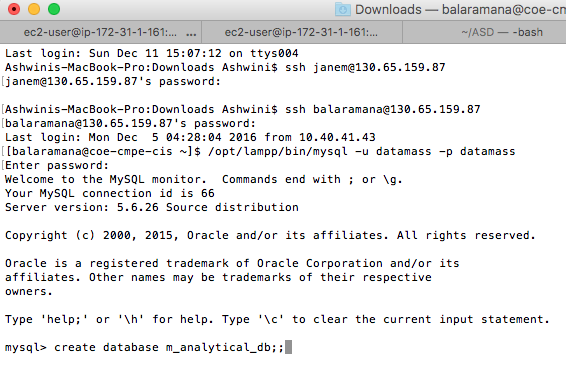
**Connect to the CIS hosted server**



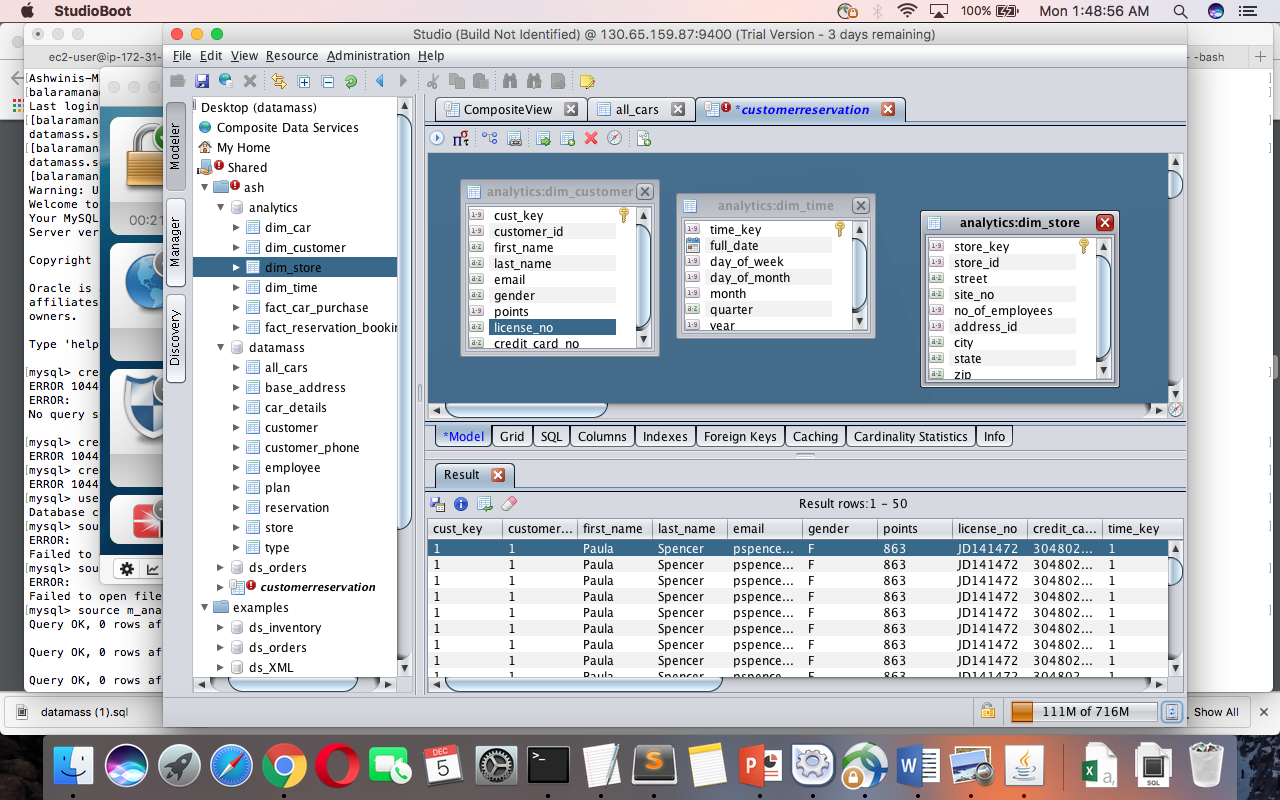
**Start the MySQL server and import the Operation Database**



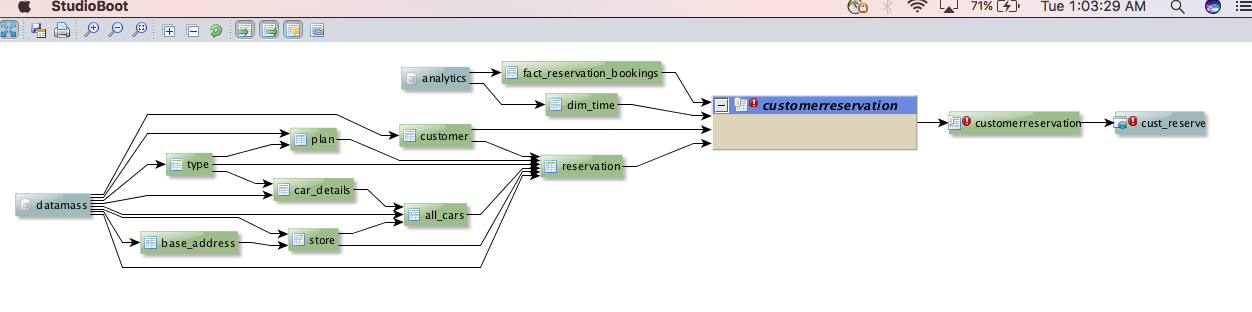
**Import the Analytical Database**



**Create a view in the CIS dashboard by drag n drop of tables you want to join and create analytics**



**Explore CIS - Find lineage - the relationship between tables in operational db and analytical db**



**9.2 Operations**

The service provides views for analytics. Finally publish the service that can be used by users with permissions. We could not publish the service and did not go beyond this point.