AMUSEMENT PARK

CS 6360.002

Team members
DEVANSHU DIPAK SHETH - dds160030
MICHAEL CHEN - mwc150230
SHALINI HEMACHANDRAN - sxh163230

Foreword

This project was a combined effort of ours and we are glad to present it to everyone. We chose this project because we felt we could do something that would inspire us to be creative as well as appeal to the inner children amongst us all.

This project is our interpretation of how an amusement park database might look like. The Amusement Park database consists of everything one would expect in a real-world amusement park such as rides, staff, visitors, gift stalls, restaurants etc.

We would like to thank everyone who supported us and a special thank you to Dr. Nurcan Yuruk, for guiding us through this project. Her helpful tips and suggestions have made this project what it is today.

Thank you, and we hope you like it.

Devanshu, Michael and Shalini

Data Requirements

RIDE

1. The amusement park has multiple rides. Each ride is uniquely identified by an ID and has a Name, Status (Functioning, Closed, Under Maintenance), maximum occupancy, duration, operating days, operating hours, Ride Type (Water or Land) and a location.
2. The rides can be either – water rides or land rides. Each water ride a water depth and each land ride has a thrill level.
3. Each ride must *have* a Ride restriction represented by an ID and the ride restriction outlines the Height restriction, Weight Restriction, and the Health requirements for a given ride. A ride may have a restriction or no restriction. The ride restrictions are generic, meaning we allow more than one ride to have same set of ride restrictions.
4. A ride *needs* ride equipment. Each ride equipment is identified by a unique ID and has a manufacturer name, cost of equipment, next maintenance date and last maintenance date. A ride must have ride equipment.

EMPLOYEE

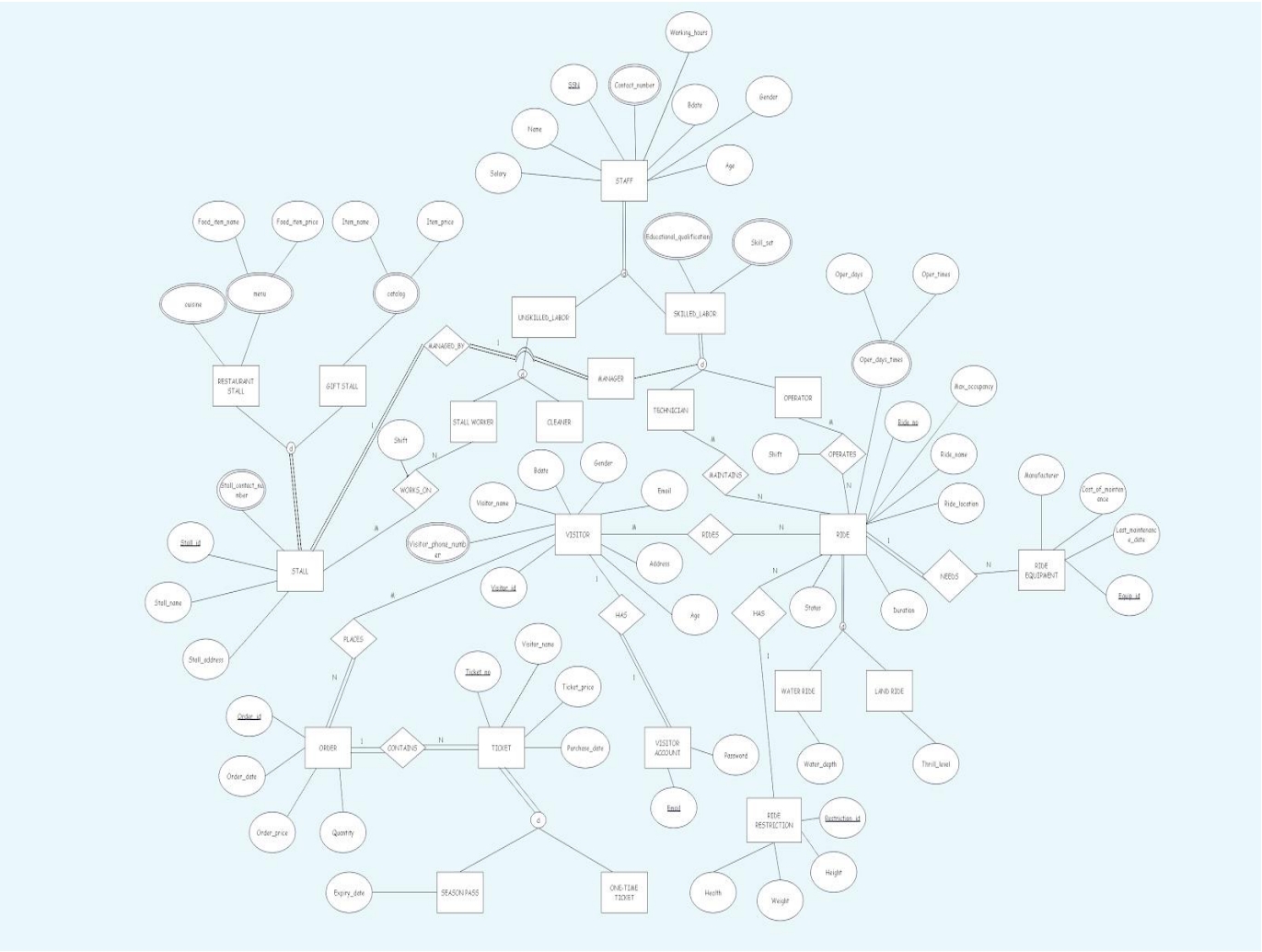
1. The amusement park has many employees, classified under Staff. The staff can be either, Skilled Labor or Unskilled Labor. We keep a record of the employee's SSN which uniquely identifies him. S/he has a Name, Salary, Contact Number (may have more than one phone number), Working Hours, Birthdate, Gender, Age and Job Type (Manager, Technician, Operator, Cleaner, Stall Worker).
2. Skilled labor includes Managers, Technicians and Operators. They are all exclusive of each other. The Skilled laborers must have a minimum educational qualification and skill set. The skill set is a list of skills of the employee, which is a multi-valued attribute.
3. A technician *maintains* a ride from time to time. A ride may or may not need a technician to maintain it. A ride can have many technicians and a technician may maintain many rides.
4. An operator *operates* a ride. A ride may or may not have an operator. Each operator has a shift which is the number of hours he works on any given ride. A ride may have multiple operators and an operator may work on many rides.
5. Unskilled labor includes stall workers and cleaners.
6. Stall workers *work on* a Stall. A stall is uniquely identified by its ID. We keep a track of the Stall Name, Location, contact number(s) and Stall Type (Restaurant or Gift) for the stall.
7. A stall is *managed by* a manager. A stall must have a manager and a manager must manage a stall.
8. A stall may be a restaurant stall or gift stall.
9. A restaurant stall may have multiple cuisines and has a menu, which will have the name of the food items available and their individual prices.

10. A gift stall has a catalog which lists the items sold by the stall and their individual prices.

VISITOR

1. A visitor visits the Amusement Park. A visitor *rides* a ride. A visitor may or may not ride a given ride. A ride may or may not have a visitor.
2. A visitor is uniquely identified by a Visitor ID. We also keep a track of the Visitor's Name, Birthdate, Gender, Email, Address, Phone Number(s) and Age.
3. A visitor may or may not have a Visitor Account for online purchases.
4. A Visitor Account is uniquely associated with an email address. Each account has a password. The email address serves as the user name.
5. A visitor may or may not *place* an Order. An order must have a Visitor associated with it.
6. Order is uniquely identified by an Order ID. The order has a Price, Quantity which is the number of tickets and the Order Date.
7. Order must *contain* a Ticket. A ticket must also be placed using an order. A ticket is uniquely identified by a Ticket Number. We also keep a track of the Visitor Name, the Ticket Price, Ticket Type (Season Pass or One-Time Ticket) and the Purchase Date.
8. A ticket may be either a Season Pass or a One-Time Ticket. If it is a Season Pass, it has an expiry date.

EER Diagram Overview



Strong Entities

- RIDE
- RIDE EQUIPMENT
- RIDE RESTRICTIONS
- CUSTOMER
- CUSTOMER ACCOUNT
- TICKET
- ORDER
- STALL
- STAFF

Specializations

- STAFF
    - SKILLED LABOR
    - UNSKILLED LABOR

- SKILLED LABOR
    - MANAGER
    - TECHNICIAN
    - OPERATOR

- UNSKILLED LABOR
    - STALL WORKER
    - CLEANER

- STALL
    - RESTAURANT STALL
    - GIFT STALL

- RIDE
    - LAND
    - WATER

Relations

One-To-One

1. HAS: VISITOR - VISITOR_ACCOUNT
2. MANAGED_BY: STALL - STALL_MANAGER

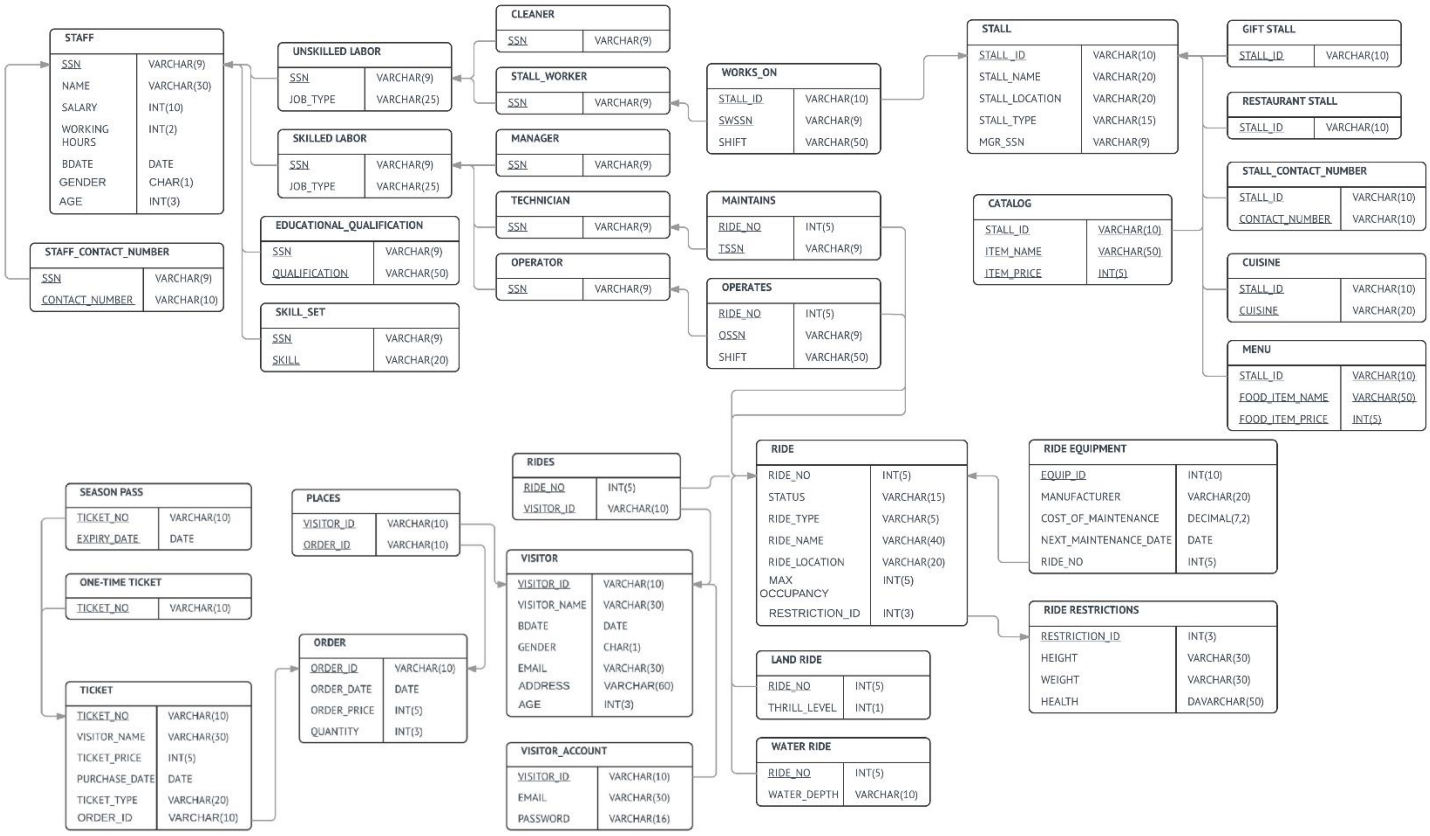
One-To-Many

1. CONTAINS: ORDER - TICKET
2. HAS: RIDE - RIDE_RESTRICTION
3. NEEDS: RIDE - RIDE_EQUIPMENTS

Many-To-Many

1. WORKS_ON: STALL_WORKER - STALL
2. MAINTAINS: TECHNICIAN -RIDE
3. OPERATES: OPERATOR - RIDE
4. RIDES: VISITOR - RIDE
5. PLACES: VISITOR - ORDER

Mapping to Relational Schema



Functional Dependencies

- Table 1: Bdate → Age

- Table 3: Quantity, Order_date → Order_price

  The thinking here is on a given day the price for a ticket may be same and that ticket quantity and order date determines the order price.

- Table 4: Purchase_date, Ticket_type → Ticket_price

  The thinking here is on a given day the price for a ticket may be same and that ticket quantity and order date determines the ticket price.

- Table 5: Bdate → Age

- Table 6: Visitor_id → Email

  Unique email ID is assumed for a customer account.

- Table 27: Food_item_name → Food_item_price

- Table 28: Item_name → Item_price

Tables

1. Staff (Salary, Name, SSN, Working_hours, Bdate, Gender, Age)
2. Stall (Stall_id, Stall_name, Stall_location, Stall_type, Mgr_ssn(FK))
3. Order (Order_id, Order_date, Order_price, Quantity)
4. Ticket (Ticket_no, Visitor_name, Ticket_price, Purchase_date, Ticket_type, Order_id (FK))
5. Visitor (Visitor_id, Visitor_name, Bdate, Gender, Email, Address, Age)
6. Visitor Account (Email, Password, Visitor_id(FK))
7. RideRestrictions (Restriction_id, Height, Weight, Health)
8. RideEquipment (Equip_id, Manufacturer, Cost_of_maintenance, Last_maintenance_date, Ride_no (FK))
9. Ride (Ride_no, Status, Ride_name, Ride_location, Max_occupancy, Duration, Ride_type, Restriction_id (FK))
10. Works_On (Stall_id (FK), SWSSN(FK), Shift)
11. Maintains (Ride_No(FK), TSSN(FK))
12. Operates (Ride_No(FK), OSSN(FK), Shift)
13. Rides (Ride_no(FK), Visitor_id(FK))
14. Places (Visiter_id(FK), Order_id(FK))
15. Staff_Contact_Number (SSN(FK), Contact_number)
16. Stall_Contact_Number (Stall_id(FK), Stall_contact_number)
17. Visitor_Contact_Number (Vistor_id(FK), Visitor_contact_number)
18. Water_Ride (Ride_no(FK), water_depth)
19. Land_Ride (Ride_no(FK), Thrill_level)
20. Season_Pass (Ticket_no(FK), Expiry_date)
21. One-Time_Ticket (Ticket_no(FK))
22. Restaurant_Stall (Stall_id(FK))
23. Gift_Stall (Stall_id(FK))
24. Skilled_Labor (SSN(FK), jobType)
25. UnSkilled_Labor(SSN(FK), jobType)
26. Cuisine (Stall_id(FK), cuisine)
27. Menu (Stall_id(FK), Food_item_name, Food_item_price)
28. Catalog (Stall_id(FK), Item_name, Item_price)
29. Skill_Set(SSN(FK), skill)
30. Educational_Qualification (SSN(FK), qualification)
31. Manager(SSN(FK))
32. Staff_Salary_log(SSN(FK), Salary(FK), Log_date)

3-NF Normalized Tables

Normalization

All tables are in 1 NF.

Table 1 is in 2 NF.
To make it 3NF,

Table 1: Staff (Salary, Name, SSN(FK), Working_hours, Bdate, Gender, Job_type)
Table 31: Staff Age (Bdate, Age)

Table 3 is in 2 NF.
To make it 3NF,

Table 3: Order (Order_id, Order_date, Quantity)
Table 32: Order_Price_Details (Order_date, Quantity, Order_price)

Table 4 is in 2 NF.
To make it 3NF,

Table 4: Ticket (Ticket_no(FK), Visitor_name, Purchase_date, Ticket_type, Order_id (FK))
Table 33: Ticket_Price_Details (Purchase_date(FK), Ticket_type(FK), Ticket_price)

Table 5 is in 2 NF.
To make it 3NF,

Table 5: Visitor (Visitor_id, Visitor_name, Bdate, Gender, Email, Address)
Table 34: Visitor Age (Bdate, Age)

All other tables are in 3 NF

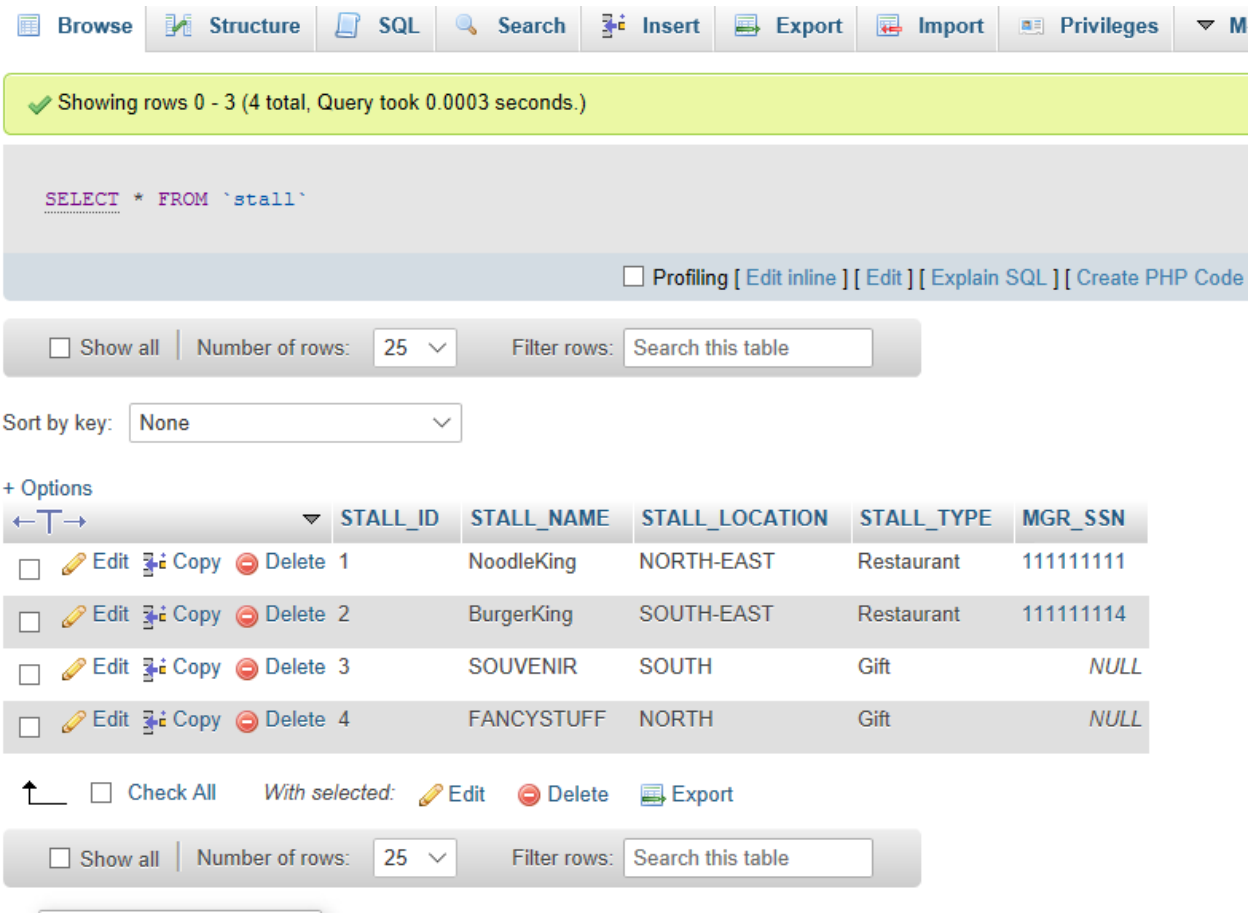
Final Tables after Normalization

1. Staff (Salary, Name, SSN, Working_hours, Bdate, Gender)
2. Stall (Stall_id, Stall_name, Stall_location, Stall_type)
3. Orders (Order_id, Order_date, Quantity)
4. Ticket (Ticket_no, Visitor_name, Purchase_date, Ticket_type, Order_id (FK))
5. Visitor (Visitor_id, Visitor_name, Bdate, Gender, Email, Address)
6. Visitor Account (Email(FK), Password, Visitor_id(FK))
7. RideRestrictions (Restriction_id, Height, Weight, Health)
8. RideEquipment (Equip_id, Manufacturer, Cost_of_maintenance, Last_maintenance_date, Ride_no (FK))
9. Ride (Ride_no, Status, Ride_name, Ride_location, Max_occupancy, Duration, Ride_type, Restriction_id (FK))
10. Works_On (Stall_id(FK), SWSSN(FK), Shift)
11. Maintains (Ride_No(FK), TSSN(FK)
12. Operates (Ride_No(FK), OSSN(FK), Shift)
13. Rides (Ride_no(FK), Visitor_id(FK))
14. Places (Visiter_id(FK), Order_id(FK))
15. Staff_Contact_Number (SSN(FK), Contact_number)
16. Stall_Contact_Number (Stall_id(FK), Stall_contact_number)
17. Visitor_Contact_Number (Vistor_id(FK), Visitor_contact_number)
18. Water_Ride (Ride_no(FK), water_depth)
19. Land_Ride (Ride_no(FK), Thrill_level)
20. Season_Pass (Ticket_no(FK), Expiry_date)
21. One-Time_Ticket (Ticket_no(FK))
22. Restaurant_Stall (Stall_id, (FK))
23. Gift_Stall (Stall_id, (FK))
24. Skilled_Labor (SSN(FK), jobType)
25. UnSkilled_Labor (SSN(FK), jobType)
26. Cuisine (Stall_id(FK), cuisine)
27. Menu (Stall_id(FK), Food_item_name, Food_item_price)
28. Catalog (Stall_id(FK), Item_name, Item_price)
29. Skill_Set (SSN(FK), skill)
30. Educational_Qualification (SSN(FK), qualification)
31. Staff Age (Bdate(FK), Age)
32. Order_Price_Details (Order_date(FK), Quantity(FK), Order_price)
33. Ticket_Price_Details (Purchase_date(FK), Ticket_type(FK), Ticket_price)
34. Visitor Age (Bdate(FK), Age)
35. Manager(SSN(FK))
36. Staff_Salary_log (SSN(FK), Salary, Log_date)
37. Manager_Start_Date (Mgr_ssn(FK), Stall_id(FK), Mgr_Start_Date)

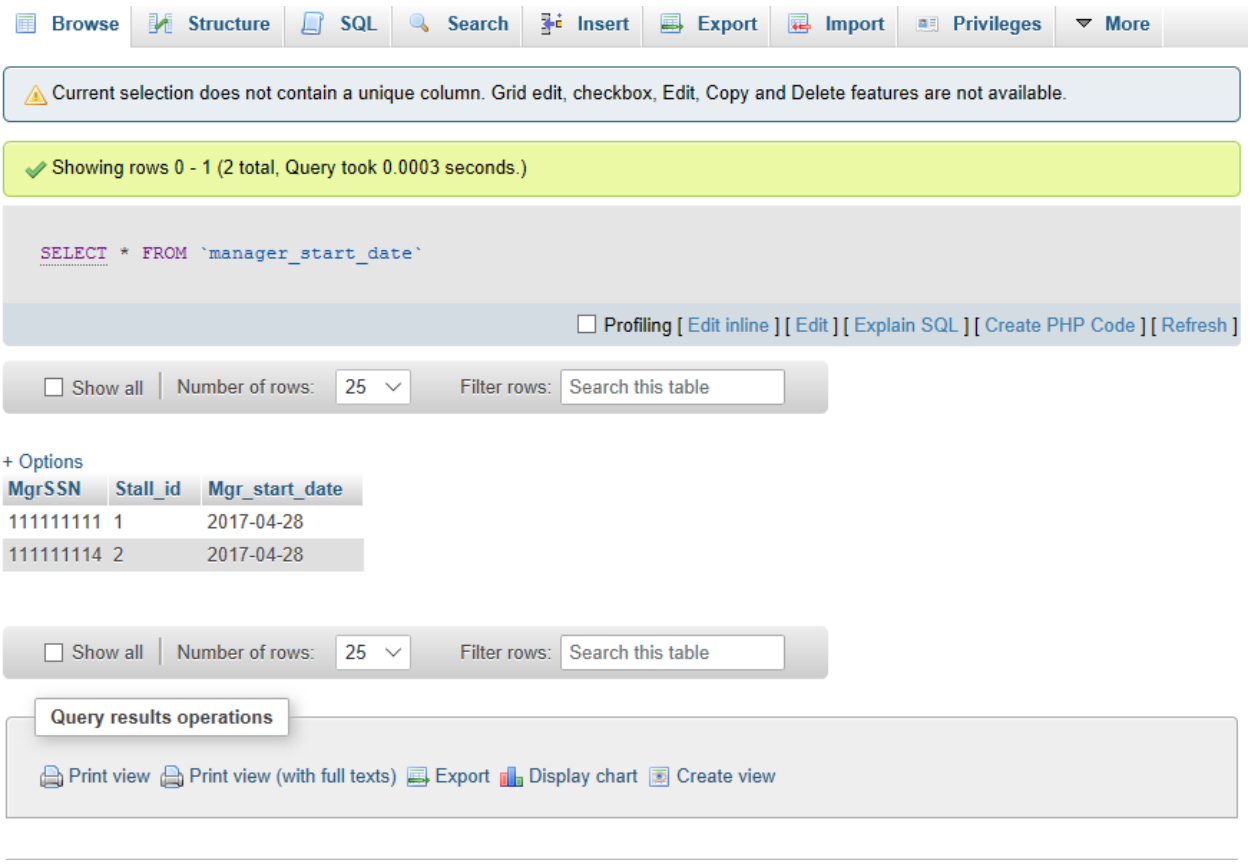
Triggers

Trigger 1: log_manager_changes

- Every time a new manager is assigned to a stall, a new entry is added in *manager_start_date* table.
- The table *manager_start_date* contains a stall id, its manager's ssn and a date which represents the date when the manager starts managing the stall.
- At any time, the table gives the date when a manager change happens with the stall.



Added Managers to NoodleKing and BurgerKing.

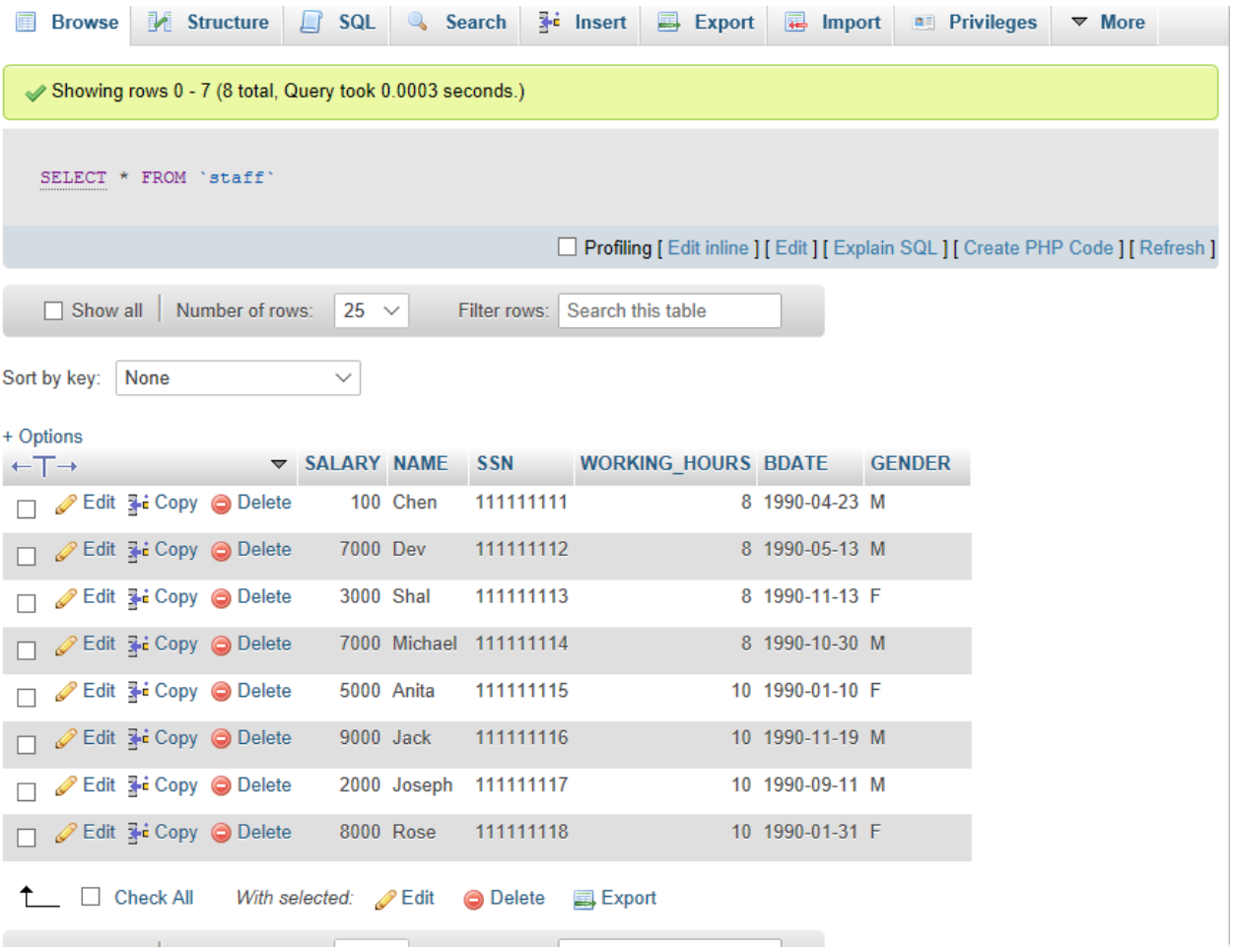


Added Managers to NoodleKing and BurgerKing.

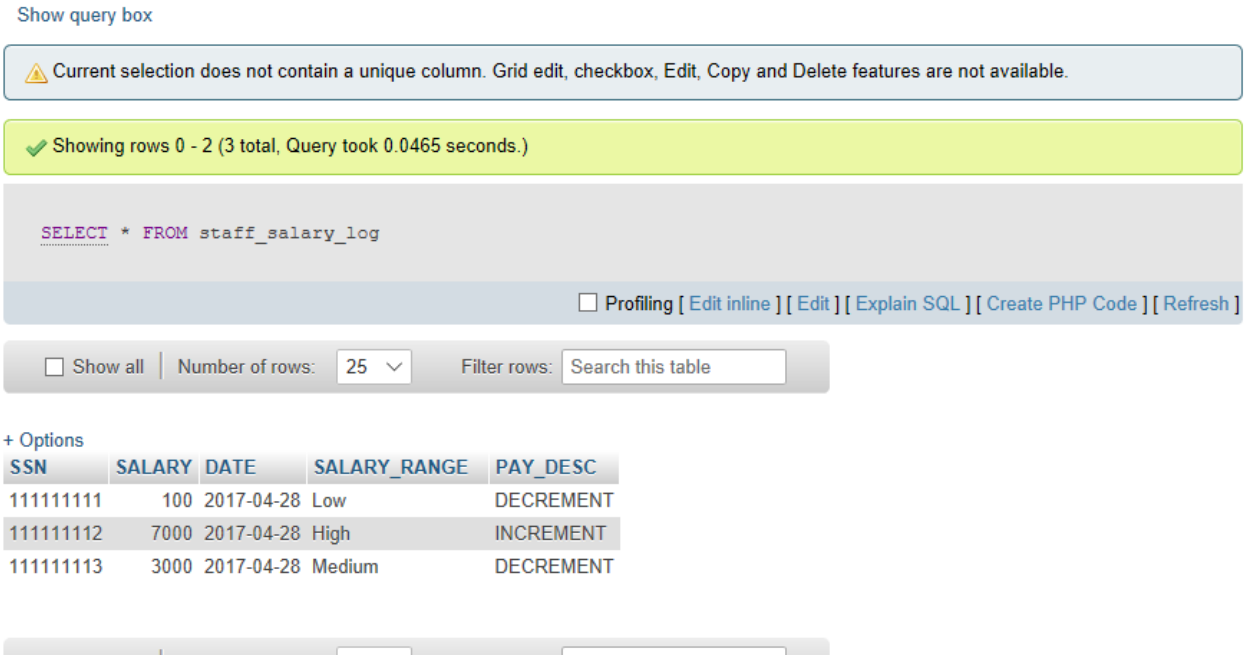
Trigger 2: log_salary_increase

- Every time an update is made on the salary field in the Staff table, a row is inserted in the *staff_salary_log table.*

- The table *staff_salary_log* contains an ssn, salary (employee's salary, date (date when the salary was updated), salary range (classification of the range of the salary) and a pay description which describes if the employee's salary increased or decreased.



Updated salaries for 1st three employees, to fit each salary range.



SSN	SALARY	DATE	SALARY_RANGE	PAY_DESC
111111111	100	2017-04-28	Low	DECREMENT
111111112	7000	2017-04-28	High	INCREMENT
111111113	3000	2017-04-28	Medium	DECREMENT

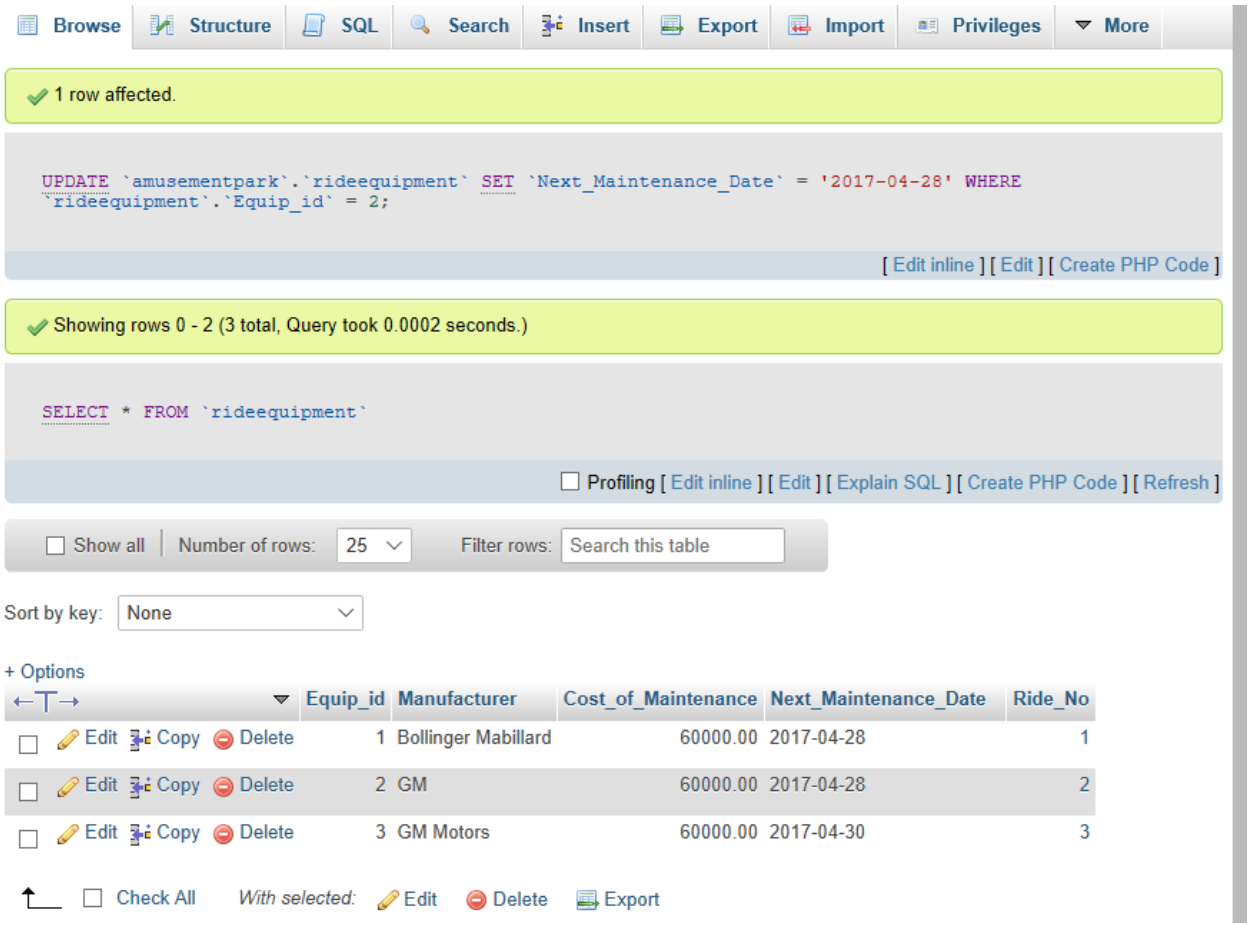
Updated salaries for 1st three employees, to fit each salary range.

Procedures

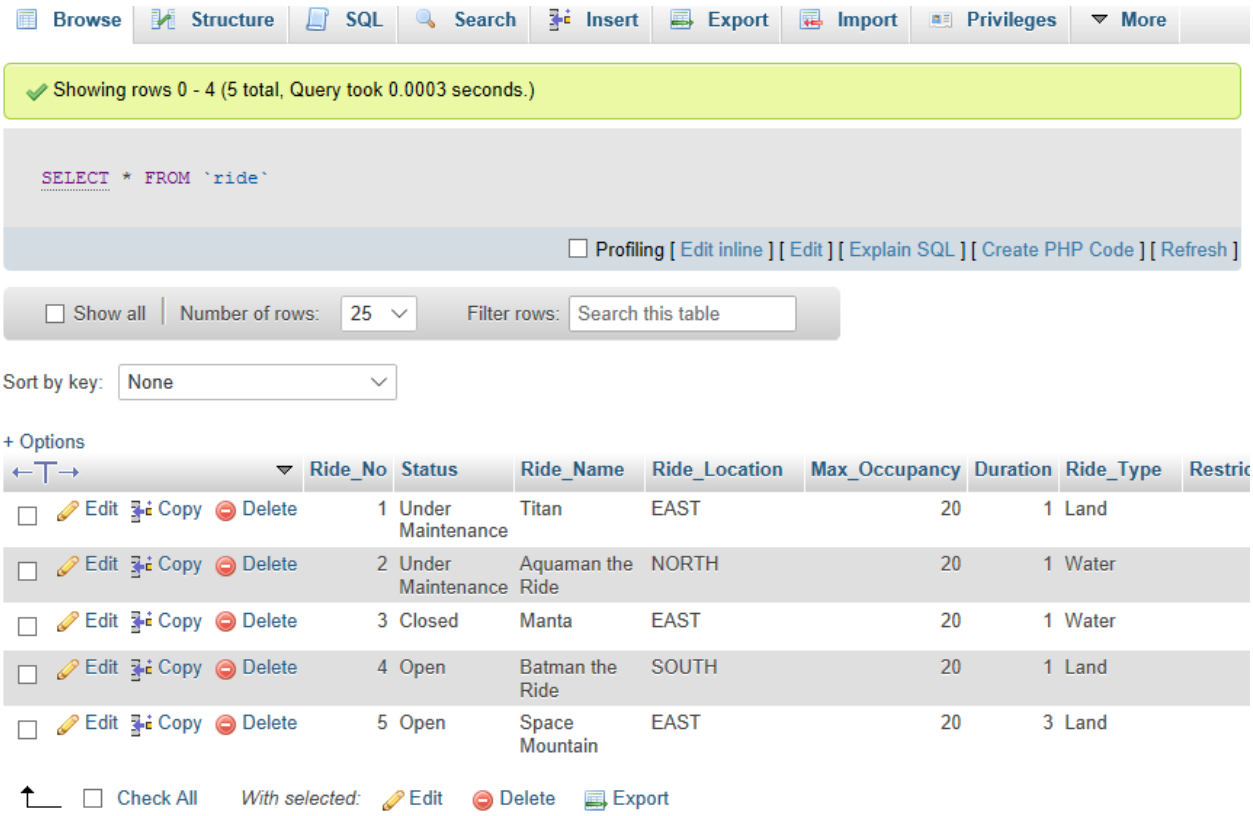
Procedure 1: maintenance_due

- The Ride_Equipment table has a next maintenance date attribute. Every ride equipment which is due for maintenance today and the rides which use this equipment are obtained.

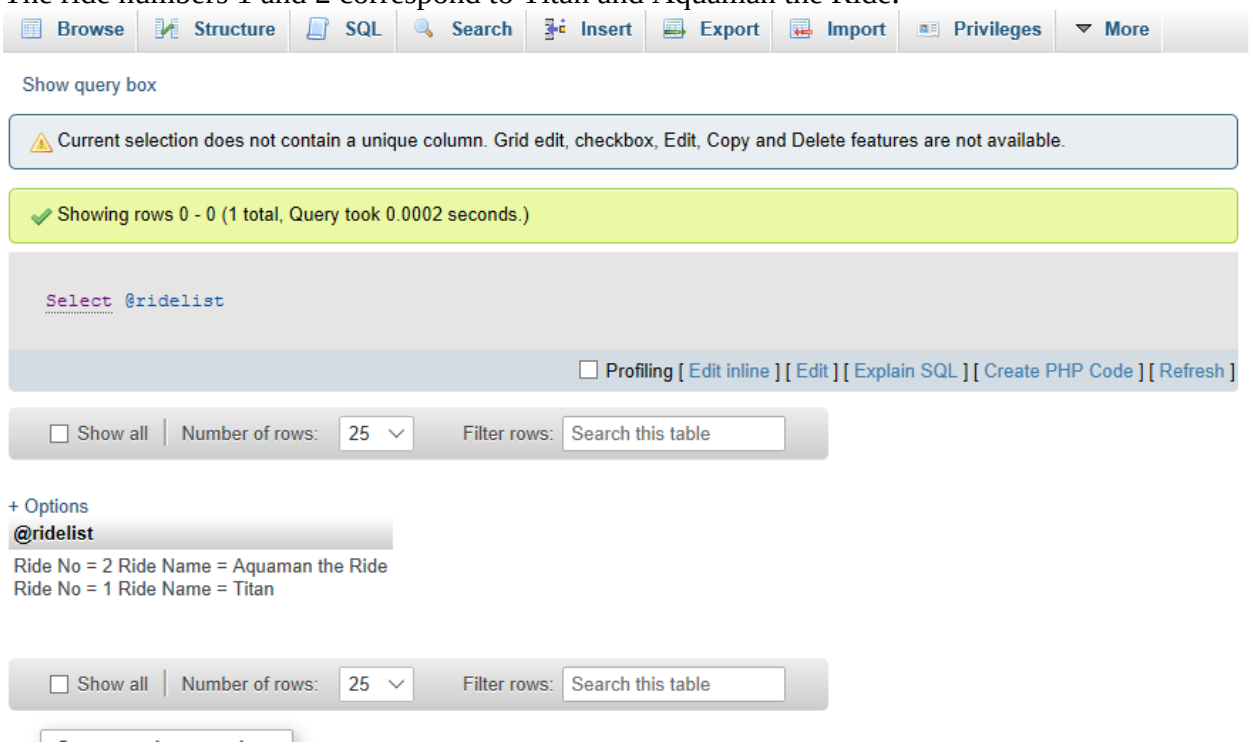
- The procedure, once executed, lists the ride numbers and names of all the rides whose maintenance date is today.



Ride equipment 1 and 2 are due today for maintenance. The rides using them are 1 and 2.



The ride numbers 1 and 2 correspond to Titan and Aquaman the Ride.

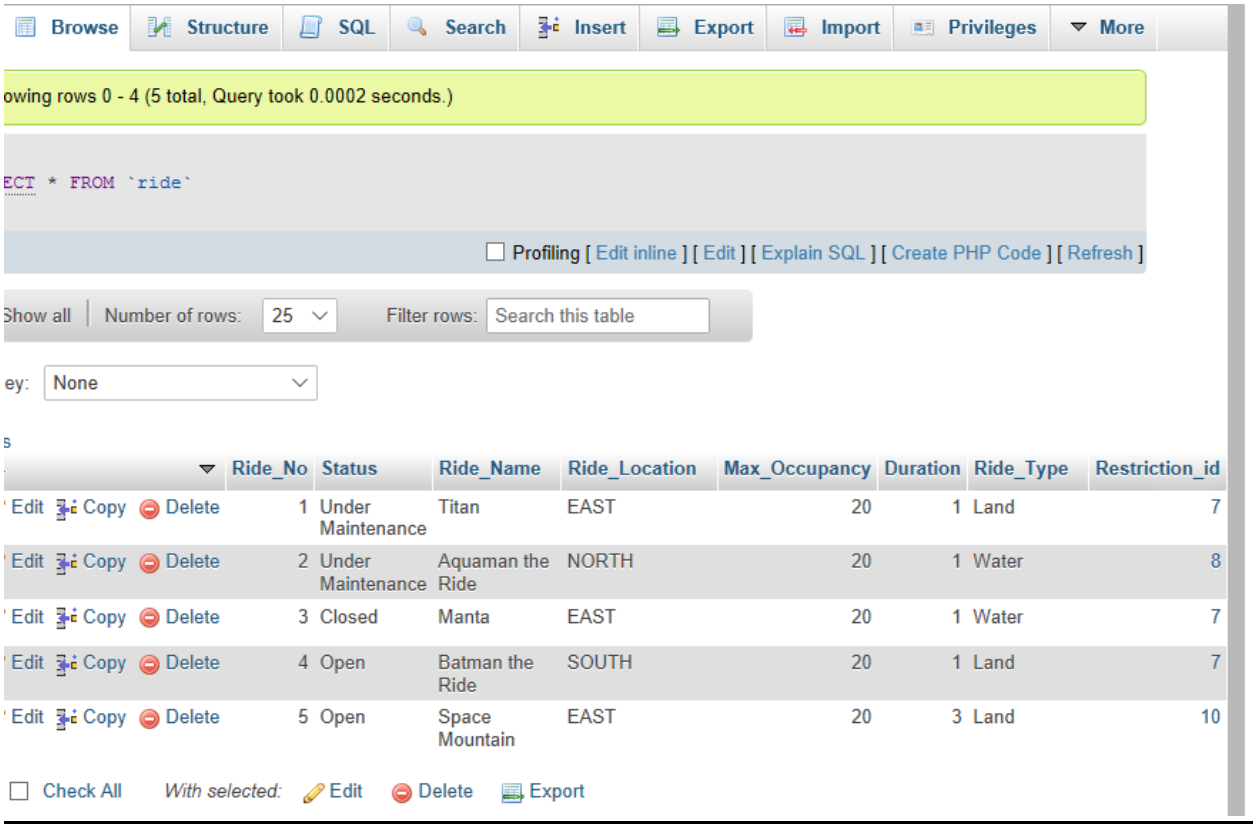


The Titan and Aquaman the Ride rides are displayed as a result.

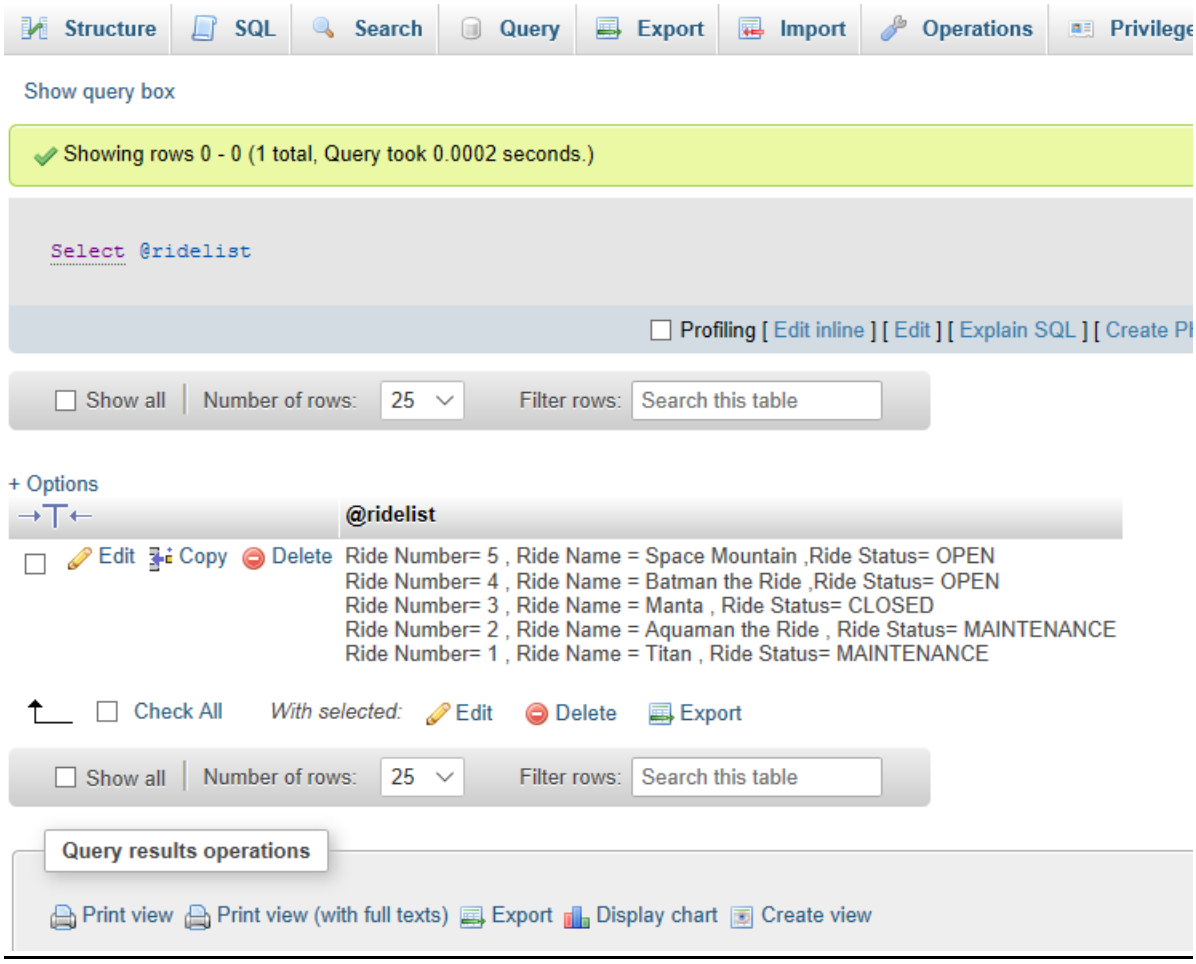
Procedure 2: ride_status

- The Ride table has an attribute called Status, which at any time, gives us the status of the ride, i.e. if it is open, closed or under maintenance.

- The procedure, once executed, lists the ride number, ride names and status of each ride in the park.



Status of each individual ride in the park.



Display status of each ride in the park with their names and numbers.

Screenshots & Source Code

Attached in zip files with project report.

Our CRUD operations have been implemented on Ride Table.

Instructions to execute the project

1. Copy the contents htdocs folder into the htdocs folder in the MAMP directory on your machine.
2. Start MAMP server
3. Execute the create table queries given in create_table_queries.txt in the sourcecodes folder
4. Execute the insert queries given in commands_for_triggers_procedures_testing.txt in the sourcecodes folder.
5. Run http://localhost/amusementparrk.html on your web browser.
6. Add rides using the form.
7. Execute triggers given in trigger1.txt and trigger2.txt in the sourcecodes folder.
8. Commands to test triggers are provided in commands_for_triggers_procedures_testing.txt in the sourcecodes folder.
9. Execute procedures given in procedure1.txt and procedure2.txt in the sourcecodes folder.
10. Calling functions for procedures are provided in the respective procedure text files.

THANK YOU