

# SYRACUSE UNIVERSITY DOME CONCESSIONS SHIFT & PAY PORTAL



**Hitesh Chandrakumar Thadhani**

**Syracuse University**

**School of Information Studies**

**IST659 – Project Implementation Report**

**Dome Concessions Shift & Pay Portal**

## INDEX

**Project Summary ----- 3**

**Tables & Attributes ----- 4**

**Entity Relationship Diagram (In Visio) ----- 7**

**Relationship Diagram (MS Access) ----- 8**

**Business Rules ----- 8**

**Database Infrastructure ----- 9**

**SQL Scripts for Creating Tables and Inserting Sample Data----- 9**

**Major Data Questions ----- 20**

**Reports ----- 21**

**Forms ----- 25**

**Procedure ----- 40**

**Trigger ----- 41**

## Project Summary:

The project proposal is about designing a database for Syracuse University Dome Concessions Organization which hosts many matches during the entire year including football, basketball. The Dome when hosting matches needs employees to take care of hospitality for the customer coming in to watch the game like food, drinks etc. Here the project is to build a database to make the schedules for the employees working for a game online through the portal.

The database would store information about employees, their schedules, managers, games and their details. It will also have details of employees who dropped shifts and who picked shifts along with their payment, hours clocked-in and their feedback from managers/supervisors. This information can be used to find number of employees working, payments made for a particular game.

The main problem with the current system is that for each game separate mail is sent to all the employees and those who respond are scheduled for that particular game. There are many instances of double scheduling, a lot of manual work to make schedule for each game. There is no automation like a portal to handle all the scheduling of games, employee schedules and payment etc.

To avoid these issues, a database design is proposed which takes care of scheduling of games and the employees that work for the games. This system can be used by employees to pick the shifts and drop them as required. Manual time and effort is saved in creating schedules for the employees and keep their track. This will eliminate the mail usage for each individual to reply to pick up a shift and trail mails for all those who want to work. The managers can just enter the shifts required for a game and all employees can pick shifts after they are posted. The managers can find out the required employees from the past records stored in database and how much is the growth in customers coming to watch games every year looking at the historical numbers in database.

The report has sections which include Entities & Attributes which talks about the various entities involved their description, primary keys, foreign keys, ERD showing how the database is designed along with business rules and some major questions which can be answered by using this system in place rather than just manual records.

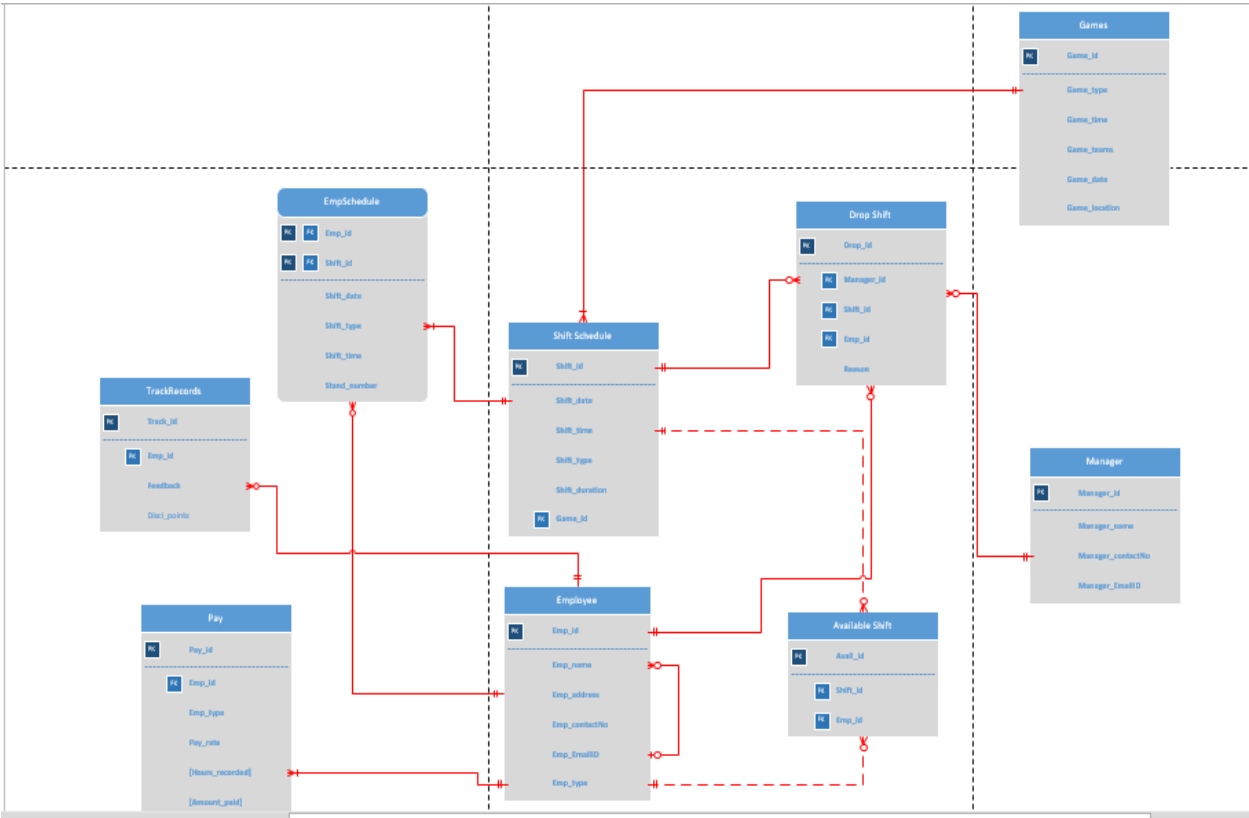
## Entities & Attributes:

Objects	Description
1. Employee	Employees working at Dome
a. Emp_id	Primary Key uniquely identifies Employee
b. Emp_name	Name of the employee
c. Emp_address	Address of the employee
d. Emp_contactNo	Contact Number of the employee
e. Emp_emailID	Email ID of the employee
f. Emp_type	Type of the employee Normal or Full Timer or Supervisor
2. Manager	Manager working for Dome
a. Manager_id	Primary Key uniquely identifies Manager
b. Manager_name	Name of the Manager
c. Manager_contactNo	Contact Number of the Manager
d. Manager_emailID	Email ID of the manager
3. Games	Schedule of the games to be played at Dome
a. Game_id	Primary Key uniquely identifies the Game
b. Game_type	Type of the Game like football, basketball etc.
c. Game_time	Time of the Game Start
d. Game_teams	Teams involved in the Game
e. Game_date	Date of the Game Played
f. Game_location	Location of the game to be played
4. Shift Schedule	Shift Details

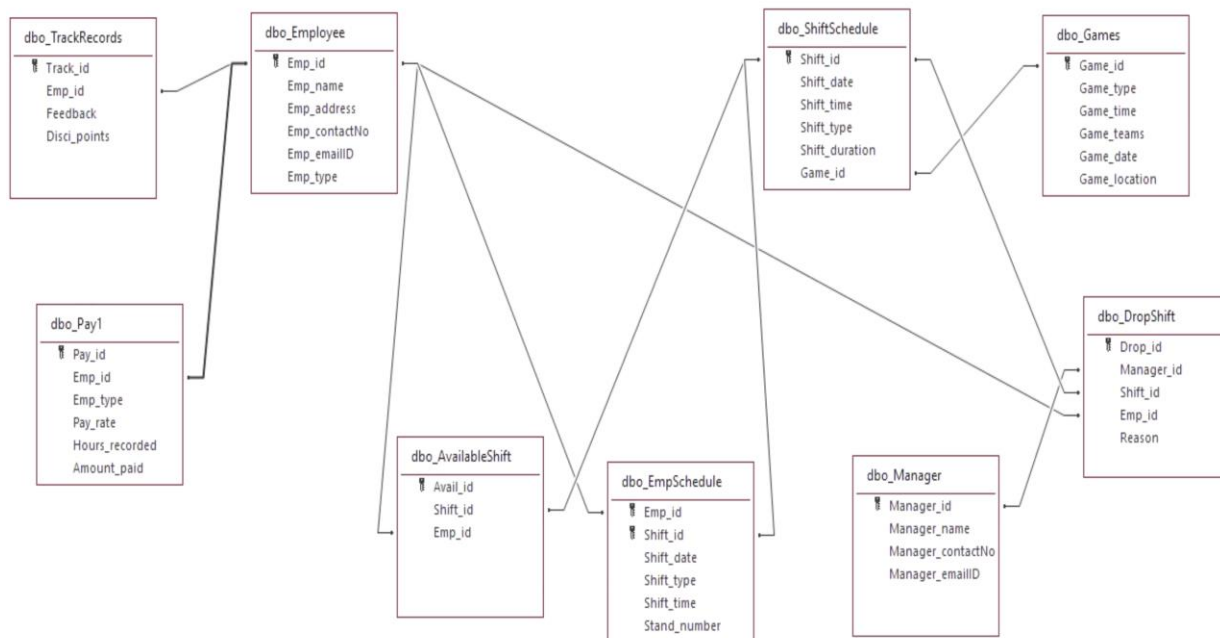
a. Shift_id	Primary Key uniquely identifies the Shift
b. Shift_date	Date of the shift
c. Shift_time	Timing of shift start
d. Shift_type	Type of the shift like Beer, Food, Portables
e. Shift_duration	Duration of the shift
f. Game_id	Foreign Key uniquely identifies Games
5. EmpSchedule	Schedule of the Employee
a. Emp_id	Primary Key, Foreign Key uniquely identifies Employee
b. Shift_id	Primary Key, Foreign Key uniquely identifies Shift
c. Shift_date	Foreign Key identifies Date of the shift
d. Shift_type	Foreign Key identifies Type of the shift
e. Shift_time	Foreign Key identifies Time of the shift start
f. Stand_number	Stand number where employee works in Dome for that particular game
6. Available Shift	Shifts that are available or picked
a. Avail_id	Primary Key uniquely identifies the shifts available or picked up
b. Shift_id	Foreign Key identifies the shift
c. Emp_id	Foreign Key identifies the employee
7. Drop Shift	Shift Dropped by Employee
a. Drop_id	Primary Key uniquely identifies shift being dropped
b. Manager_id	Foreign Key uniquely identifies the Manager for the Shift approval

c. Shift_id	Foreign Key uniquely identifies Shift
d. Emp_id	Foreign Key identifies the employee
e. Reason	Reason why the employee dropped the shift
8. Payment	Payment of employee
a. Pay_id	Payment ID uniquely identifies Payment
b. Emp_id	Foreign Key uniquely identifies Employee
c. Emp_type	Foreign Key uniquely identifies employee type
d. Pay_rate	Pay rate per hour for different employee types
e. Hours_recorded	Hours recorded for the Employee to be paid
f. Amount_paid	Amount to be paid to the employee
9. TrackRecords	Track records of the employee working at Dome
a. Track_id	Primary Key uniquely identifies Track Records
b. Emp_id	Foreign Key uniquely identifies employee
c. Feedback	Feedback given to the employee for their work
d. Disci_points	Disciplinary points given to the employee for not working properly or being late etc.

Entity Relationship Diagram (In Visio):



### Relationship Diagram (MS Access):



### Business Rules:

1. A student cannot work for more than 20 hours a week.
2. Every person who wants to login needs to be logged in as manager, supervisor, student or Administrative Team.
3. Payment can be made in cheque or direct deposit only if configured in Myslice.
4. An employee cannot drop shift 8 hours prior to the start of the game.
5. Dropping the shift needs manager's approval so that employee can drop it and it is available for pick up by another employee.
6. Shift is available after manager approval in Available shift.
7. Employee has to take 2 shifts per semester in DOME to be considered active student.
8. An employee can give feedback about other employees.
9. Manager\_id 1001 is a dummy manager which means no approval yet for drop shift.



## Database Infrastructure:

The database infrastructure uses client-server architecture model. The Microsoft SQL- Server is used as a database whereas MS Access is used as a tool of Interface. The data can be inserted, updated, deleted or be queried using the MS Access interface tool and reports can be generated along with forms to answer major data questions. Any changes done from MS Access will be reflected back to SQL Server & vice versa.

## SQL Scripts for Creating and Inserting Sample Data:

### Create Table Employee

```
CREATE TABLE Employee
(
    Emp_id NUMERIC(10,0) NOT NULL,
    Emp_name VARCHAR(20) NOT NULL,
    Emp_address VARCHAR(30) NOT NULL,
    Emp_contactNo NUMERIC(12,0) NOT NULL,
    Emp_emailID VARCHAR(20) NOT NULL,
    Emp_type VARCHAR(20) NOT NULL CHECK(Emp_type IN('Normal','Supervisor','Full time')),

    CONSTRAINT Emp_Pk PRIMARY KEY (Emp_id)
);
```

### Create Table Games

```
CREATE TABLE Games
(
    Game_id NUMERIC(10,0) NOT NULL,
    Game_type VARCHAR(12) NOT NULL,
    Game_time TIME(0) NOT NULL,
    Game_teams VARCHAR(30) NOT NULL,
    Game_date DATE NOT NULL,
    Game_location VARCHAR(15)

    CONSTRAINT Game_Pk PRIMARY KEY (Game_id)
);
```

### Create Table Manager

```
CREATE TABLE Manager
(
    Manager_id NUMERIC(10,0) NOT NULL,
    Manager_name VARCHAR(12) NOT NULL,
    Manager_contactNo NUMERIC(12,0) NOT NULL,
    Manager_emailID VARCHAR(20) NOT NULL,

    CONSTRAINT Manager_Pk PRIMARY KEY (Manager_id)
);
```

### Create Table ShiftSchedule

```
CREATE TABLE ShiftSchedule
(
    Shift_id NUMERIC(10,0) NOT NULL,
    Shift_date DATE NOT NULL,
    Shift_time TIME(0) NOT NULL,
    Shift_type VARCHAR(12) NOT NULL,
    Shift_duration NUMERIC(10,0) NOT NULL,
    Game_id NUMERIC(10,0) NOT NULL,

    CONSTRAINT Shift_Pk PRIMARY KEY (Shift_id),
    CONSTRAINT Shift_Fk FOREIGN KEY(Game_id) REFERENCES Games(Game_id)
);
```

### Create Table TrackRecords

```

CREATE TABLE TrackRecords
(
    Track_id NUMERIC(10,0) NOT NULL,
    Emp_id NUMERIC(10,0) NOT NULL,
    Feedback VARCHAR(40) NOT NULL,
    Disci_points NUMERIC(10,0) NOT NULL,

    CONSTRAINT Track_Pk PRIMARY KEY (Track_id),
    CONSTRAINT Track_FK FOREIGN KEY(Emp_id) REFERENCES Employee(Emp_id)
);

```

### Create Table EmpSchedule

```

CREATE TABLE EmpSchedule
(
    Emp_id NUMERIC(10,0) NOT NULL,
    Shift_id NUMERIC(10,0) NOT NULL,
    Shift_date DATE NOT NULL,
    Shift_type VARCHAR(12) NOT NULL,
    Shift_time TIME(0) NOT NULL,
    Stand_number VARCHAR(12) NOT NULL,

    CONSTRAINT EmpSch_Pk PRIMARY KEY (Emp_id,Shift_id),
    CONSTRAINT EmpSch_FK1 FOREIGN KEY(Emp_id) REFERENCES Employee(Emp_id),
    CONSTRAINT EmpSch_FK2 FOREIGN KEY(Shift_id) REFERENCES ShiftSchedule(Shift_id)
);

```

### Create Table Pay

```

CREATE TABLE Pay
(
    Pay_id NUMERIC(10,0) NOT NULL,
    Emp_id NUMERIC(10,0) NOT NULL,
    Emp_type VARCHAR(20) NOT NULL CHECK(Emp_type IN('Normal','Supervisor','Full time')),
    Pay_rate DECIMAL(10,2) NOT NULL,
    Hours_recorded DECIMAL(10,2) NOT NULL,
    Amount_paid DECIMAL(10,2) NOT NULL,

    CONSTRAINT Pay_Pk PRIMARY KEY (Pay_id),
    CONSTRAINT Pay_FK1 FOREIGN KEY(Emp_id) REFERENCES Employee(Emp_id),
);

```

## Create Table DropShift

```
CREATE TABLE DropShift
(
    Drop_id NUMERIC(10,0) NOT NULL,
    Manager_id NUMERIC(10,0) NOT NULL,
    Shift_id NUMERIC(10,0) NOT NULL,
    Emp_id NUMERIC(10,0) NOT NULL,
    Reason VARCHAR(30) NOT NULL,

    CONSTRAINT Drop_Pk PRIMARY KEY (Drop_id),
    CONSTRAINT Drop_FK1 FOREIGN KEY(Emp_id) REFERENCES Employee(Emp_id),
    CONSTRAINT Drop_FK2 FOREIGN KEY(Shift_id) REFERENCES ShiftSchedule(Shift_id),
    CONSTRAINT Drop_FK3 FOREIGN KEY(Manager_id) REFERENCES Manager(Manager_id)
);
```

## Create Table AvailableShift

```
CREATE TABLE AvailableShift
(
    Avail_id NUMERIC(10,0) NOT NULL,
    Shift_id NUMERIC(10,0) NOT NULL,
    Emp_id NUMERIC(10,0) NOT NULL,

    CONSTRAINT Avail_Pk PRIMARY KEY (Avail_id),
    CONSTRAINT Avail_FK1 FOREIGN KEY(Emp_id) REFERENCES Employee(Emp_id),
    CONSTRAINT Avail_FK2 FOREIGN KEY(Shift_id) REFERENCES ShiftSchedule(Shift_id)
);
```

**Insert Data in Employee Table:**

```

insert into Employee values (101,'Akash','118 Concord Place',3158804123,'aks123@gmail.com','Normal')
insert into Employee values (102,'Neeraj','103 Victoria Place',3159904478,'ner290@gmail.com','Normal')
insert into Employee values (103,'Nikhil','1021 Westcott',3159907413,'nkh1027@gmail.com','Full time')
insert into Employee values (104,'Pratik','702 South Beech St',3159905615,'pra2601@gmail.com','Supervisor')
insert into Employee values (105,'Amit','525 Columbus Ave',3159904531,'amit0315@gmail.com','Full time')
insert into Employee values (106,'Bala','200 Euclid Ave',3159903210,'bala0707@gmail.com','Supervisor')
insert into Employee values (107,'Ajay','524 Euclid Ave',3159901432,'ajay0801@gmail.com','Normal')
insert into Employee values (108,'Ashutosh','525 Westcott',3159900011,'ashu1201@gmail.com','Supervisor')
insert into Employee values (109,'Sharvil','118 Concord Place',3159908074,'shar1102@gmail.com','Supervisor')
insert into Employee values (110,'Pragya','723 Westcott',3159903189,'pra2109@gmail.com','Supervisor')

```

Select \* from Employee

100 %

Results Messages

	Emp_id	Emp_name	Emp_address	Emp_contactNo	Emp_emailID	Emp_type
1	101	Akash	118 Concord Place	3158804123	aks123@gmail.com	Normal
2	102	Neeraj	103 Victoria Place	3159904478	ner290@gmail.com	Normal
3	103	Nikhil	1021 Westcott	3159907413	nkh1027@gmail.com	Full time
4	104	Pratik	702 South Beech St	3159905615	pra2601@gmail.com	Supervisor
5	105	Amit	525 Columbus Ave	3159904531	amit0315@gmail.com	Full time
6	106	Bala	200 Euclid Ave	3159903210	bala0707@gmail.com	Supervisor
7	107	Ajay	524 Euclid Ave	3159901432	ajay0801@gmail.com	Normal
8	108	Ashutosh	520 Westcott	3159900011	ashu1201@gmail.com	Supervisor
9	109	Sharvil	118 Concord Place	3159908074	shar1102@gmail.com	Supervisor
10	110	Pragya	723 Westcott	3159903189	pra2109@gmail.com	Supervisor

**Insert Data in Games Table:**

```

insert into Games values (1,'football','19:00:00','Syracuse Boston','09-07-2019','Dome')
insert into Games values (2,'basketball','14:00:00','Syracuse Bucknell','09-22-2019','Dome')
insert into Games values (3,'football','19:00:00','Syracuse Virginia','09-30-2019','Dome')
insert into Games values (4,'basketball','18:00:00','Syracuse Chicago','10-10-2019','Dome')
insert into Games values (5,'football','14:00:00','Syracuse Clemson','10-25-2019','Dome')

```

Select \* from Games

100 %

Results Messages

	Game_id	Game_type	Game_time	Game_teams	Game_date	Game_location
1	1	football	19:00:00	Syracuse Boston	2019-09-07	Dome
2	2	basketball	14:00:00	Syracuse Bucknell	2019-09-22	Dome
3	3	football	19:00:00	Syracuse Virginia	2019-09-30	Dome
4	4	basketball	18:00:00	Syracuse Chicago	2019-10-10	Dome
5	5	football	14:00:00	Syracuse Clemson	2019-10-25	Dome

### Insert Data in Manager Table:

```

insert into Manager values(10, 'Mahesh', 3159902378, 'mahe23@gmail.com')
insert into Manager values(11, 'Tanveer', 3159902351, 'tan0317@gmail.com')
insert into Manager values(12, 'Nikunj', 3159900390, 'nik0519@gmail.com')
insert into Manager values(13, 'Akshay', 3159905566, 'aks1803@gmail.com')
insert into Manager values(14, 'Rahul', 3159907373, 'rahul1508@gmail.com')
insert into Manager values(15, 'Abhijeet', 3159902056, 'abhi1208@gmail.com')
insert into Manager values(1001, 'None', 0000000000, 'NONE')

```

Select \* from Manager

100 %

Results Messages

	Manager_id	Manager_name	Manager_contactNo	Manager_emailID
1	10	Mahesh	3159902378	mahe23@gmail.com
2	11	Tanveer	3159902351	tan0317@gmail.com
3	12	Nikunj	3159900390	nik0519@gmail.com
4	13	Akshay	3159905566	aks1803@gmail.com
5	14	Rahul	3159907373	rahul1508@gmail.com
6	15	Abhijeet	3159902056	abhi1208@gmail.com
7	1001	None	0	NONE

**Insert Data in ShiftSchedule Table:**

```

insert into ShiftSchedule values (201,'09-07-2019','17:30:00','Beer',5,1)
insert into ShiftSchedule values (202,'09-07-2019','17:00:00','Food',6,1)
insert into ShiftSchedule values (203,'09-22-2019','12:30:00','Portable',4,2)
insert into ShiftSchedule values (204,'09-30-2019','18:00:00','Beer',5,3)
insert into ShiftSchedule values (205,'10-10-2019','17:00:00','Popcorn',5,4)
insert into ShiftSchedule values (206,'10-10-2019','17:00:00','Beer',5,4)
insert into ShiftSchedule values (207,'10-10-2019','17:00:00','Food',5,4)
insert into ShiftSchedule values (208,'10-25-2019','12:30:00','Food',5,5)
insert into ShiftSchedule values (209,'09-30-2019','17:30:00','Food',5,3)
insert into ShiftSchedule values (210,'09-22-2019','13:00:00','Beer',5,2)

```

Select \* from ShiftSchedule

100 %

Results Messages

	Shift_id	Shift_date	Shift_time	Shift_type	Shift_duration	Game_id
1	201	2019-09-07	17:30:00	Beer	5	1
2	202	2019-09-07	17:00:00	Food	6	1
3	203	2019-09-22	12:30:00	Portable	4	2
4	204	2019-09-30	18:00:00	Beer	5	3
5	205	2019-10-10	17:00:00	Popcom	5	4
6	206	2019-10-10	17:00:00	Beer	5	4
7	207	2019-10-10	17:00:00	Food	5	4
8	208	2019-10-25	12:30:00	Food	5	5
9	209	2019-09-30	17:30:00	Food	5	3
10	210	2019-09-22	13:00:00	Beer	5	2

**Insert Data in TrackRecords Table:**

```

insert into TrackRecords values (21,101,'ontime good',0)
insert into TrackRecords values (22,102,'late',1)
insert into TrackRecords values (23,103,'excellent',0)
insert into TrackRecords values (24,106,'needs to take less break',1)
insert into TrackRecords values (25,104,'good at handling customers',0)
insert into TrackRecords values (26,101,'good',0)
insert into TrackRecords values (27,102,'late',1)
insert into TrackRecords values (28,109,'Very Slow',2)

```

Select \* from TrackRecords

100 %

Results Messages

	Track_id	Emp_id	Feedback	Disci_points
1	21	101	ontime good	0
2	22	102	late	1
3	23	103	excellent	0
4	24	106	needs to take less break	1
5	25	104	good at handling customers	0
6	26	101	good	0
7	27	102	late	1
8	28	109	Very Slow	2

### Insert Data in EmpSchedule Table:

```

insert into EmpSchedule values (101,201,'09-07-2019','Beer','17:30:00','147B')
insert into EmpSchedule values (101,204,'09-30-2019','Beer','18:00:00','142')
insert into EmpSchedule values (101,205,'10-10-2019','Popcorn','17:00:00','247B')
insert into EmpSchedule values (102,202,'09-07-2019','Food','17:00:00','126A')
insert into EmpSchedule values (102,203,'09-22-2019','Portable','12:30:00','270B')
insert into EmpSchedule values (103,204,'09-30-2019','Beer','17:00:00','142A')
insert into EmpSchedule values (103,205,'10-10-2019','Popcorn','17:00:00','270C')
insert into EmpSchedule values (104,204,'10-10-2019','Beer','17:00:00','147C')
insert into EmpSchedule values (104,209,'09-30-2019','Food','17:30:00','252D')
insert into EmpSchedule values (105,210,'09-22-2019','Beer','13:00:00','280D')
insert into EmpSchedule values (106,203,'09-22-2019','Portable','12:30:00','264B')
insert into EmpSchedule values (107,208,'10-25-2019','Food','12:30:00','264C')
insert into EmpSchedule values (108,209,'09-30-2019','Food','17:30:00','260C')
insert into EmpSchedule values (109,206,'10-10-2019','Beer','17:00:00','271C')
insert into EmpSchedule values (110,202,'09-07-2019','Food','17:00:00','210D')

```



Select \* from EmpSchedule

100 %

Results Messages

	Emp_id	Shift_id	Shift_date	Shift_type	Shift_time	Stand_number
1	101	201	2019-09-07	Beer	17:30:00	147B
2	101	204	2019-09-30	Beer	18:00:00	142
3	101	205	2019-10-10	Popcom	17:00:00	247B
4	102	202	2019-09-07	Food	17:00:00	126A
5	102	203	2019-09-22	Portable	12:30:00	270B
6	103	204	2019-09-30	Beer	18:00:00	142A
7	103	205	2019-10-10	Popcom	17:00:00	270C
8	104	204	2019-09-30	Beer	18:00:00	147C
9	104	208	2019-10-25	Food	12:30:00	252D
10	105	210	2019-09-22	Beer	13:00:00	280D
11	106	203	2019-09-22	Portable	12:30:00	264B
12	107	208	2019-10-25	Food	12:30:00	264C
13	108	209	2019-09-30	Food	17:30:00	260C
14	109	206	2019-10-10	Beer	17:00:00	271C
15	110	202	2019-09-07	Food	17:00:00	210D

✓ Query executed successfully.

### Insert Data in Pay Table:

```

insert into Pay values (41,101,'Normal',11.60,0.0,0.0)
insert into Pay values (42,102,'Normal',11.60,0.0,0.0)
insert into Pay values (43,103,'Full time',15.00,0.0,0.0)
insert into Pay values (44,104,'Supervisor',13.00,0.0,0.0)
insert into Pay values (45,106,'Supervisor',13.00,0.0,0.0)
insert into Pay values (46,107,'Normal',11.60,0.0,0.0)
insert into Pay values (47,108,'Normal',11.60,0.0,0.0)
insert into Pay values (48,109,'Supervisor',13.00,0.0,0.0)
insert into Pay values (49,110,'Supervisor',13.00,0.0,0.0)
INSERT INTO Pay Values (50,105,'Full time',15.00,0.00,0.00)

```

Select \* from Pay

100 %

Results Messages

	Pay_id	Emp_id	Emp_type	Pay_rate	Hours_recorded	Amount_paid
1	41	101	Normal	11.60	0.00	0.00
2	42	102	Normal	11.60	0.00	0.00
3	43	103	Full time	15.00	0.00	0.00
4	44	104	Supervisor	13.00	0.00	0.00
5	45	106	Supervisor	13.00	0.00	0.00
6	46	107	Normal	11.60	0.00	0.00
7	47	108	Normal	11.60	0.00	0.00
8	48	109	Supervisor	13.00	0.00	0.00
9	49	110	Supervisor	13.00	0.00	0.00
10	50	105	Full time	15.00	0.00	0.00

### Insert Data in DropShift Table:

```

insert into DropShift values(51,1001,204,101,'Exceeding hours')
insert into DropShift values(52,1001,210,105,'Out of town')
insert into DropShift values(53,1001,204,104,'Exceeding hours')
insert into DropShift values(54,1001,209,108,'Exceeding hours')
insert into DropShift values(55,1001,208,107,'Exceeding hours')
insert into DropShift values(56,1001,204,103,'Sick')

```

Select \* from DropShift

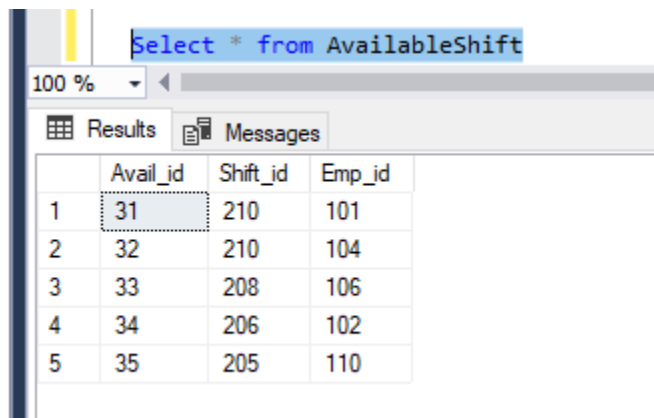
100 %

Results Messages

	Drop_id	Manager_id	Shift_id	Emp_id	Reason
1	51	1001	204	101	Exceeding hours
2	52	1001	210	105	Out of town
3	53	1001	204	104	Exceeding hours
4	54	1001	209	108	Exceeding hours
5	55	1001	208	107	Exceeding hours
6	56	1001	204	103	Sick

**Insert Data in AvailableShift Table:**

```
insert into AvailableShift values(31,210,101)
insert into AvailableShift values(32,210,104)
insert into AvailableShift values(33,208,106)
insert into AvailableShift values(34,206,102)
insert into AvailableShift values(35,205,110)
```



The screenshot shows a SQL Server Enterprise Manager window with the query "Select \* from AvailableShift" entered in the query editor. The results pane displays a table with 5 rows and 4 columns: Avail\_id, Shift\_id, and Emp\_id. The first row is highlighted with a dashed border.

	Avail_id	Shift_id	Emp_id
1	31	210	101
2	32	210	104
3	33	208	106
4	34	206	102
5	35	205	110

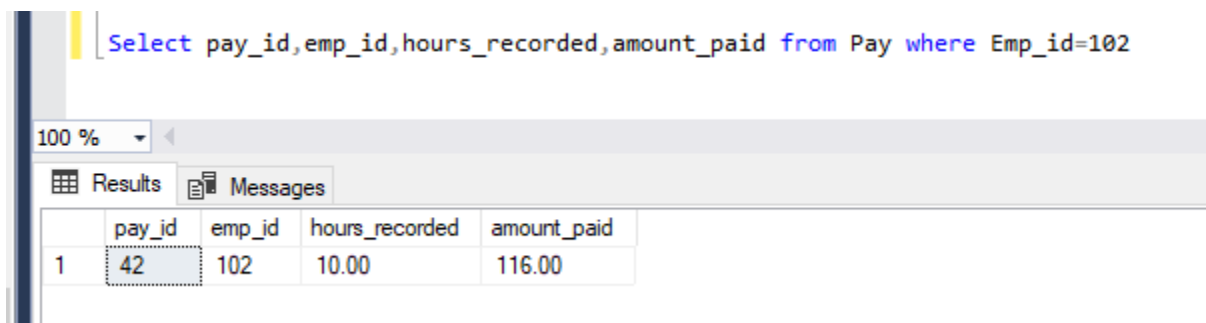
## Major Data Questions:

The Online portal is designed to eliminate the manual work going in scheduling and payroll of the employees. The database application can be a very efficient way of managing the staff and schedules online without manual intervention.

The following Major Data Questions can be asked from the database application for which I have created reports for the 3 questions where (Y) is included at the end. These reports can be directly fetched and viewed to answer those queries which are included in the next page.

- Which students pick shifts for the game and which do not? Active Students can be found out. (Y)
- How Feedback can be used to track students work and their efficiency? (Y)
- What is the average payment made for a particular game to employees? (N)
- What are the reasons given by employees to drop a shift? (Y)
- What is the annual growth in terms of customers per game? (N)

An employee can query and check how many hours he has worked along with the amount that would be paid to him by the Dome Concessions using the following query.



The screenshot shows a database query interface. At the top, a SQL query is entered: `Select pay_id,emp_id,hours_recorded,amount_paid from Pay where Emp_id=102`. Below the query, there are tabs for 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with the following data:

	pay_id	emp_id	hours_recorded	amount_paid
1	42	102	10.00	116.00

## Reports:

This database helps managers and admin persons to generate 3 reports based on the major data questions being asked.

### Report 1: Which students pick the most shifts?

Active Students working at the dome can be found out using this report.

#### SQL Query used for the report:

```
SELECT emp.Emp_id, Count(shift.Shift_duration) AS Shift_Count
FROM ShiftSchedule shift INNER JOIN (Employee emp INNER JOIN EmpSchedule ems ON emp.Emp_id = ems.Emp_id) ON shift.Shift_id = ems.Shift_id
GROUP BY emp.Emp_id;
```

Most Active Employees	
Emp_id	Shift_Count
101	3
102	2
103	2
104	2
105	1
106	1
107	1
108	1
109	1
110	1

Page: 1 | No Filter

**Report 2:** What are the most common reasons given by the employees who drop the shift?

This helps to find most frequent reasons given by the employees who drop shift and can be helpful to figure out if it a genuine reason or not.

**SQL Query used for the report:**

```
Select Reason, count(*) Reason_count from DropShift group by Reason
```

### Most\_Common\_Given\_Reasons\_DropShift

Reason	Reason_Count
Exceeding hours	4
Out of town	1
Sick	1

Monday, December 2, 2019

Page 1 of 1

**Report 3:** What is the efficiency of the employees working at the Dome?

This report helps to point out the average disciplinary points each employee gets to evaluate the efficiency of the employees.

**SQL Query used for this report:**

```
SELECT emp.Emp_id, Avg(trk.Disci_points) AS Avg_Disci_points, emp.Emp_type  
FROM Employee emp LEFT JOIN TrackRecords trk ON emp.Emp_id = trk.Emp_id  
GROUP BY emp.Emp_id, emp.Emp_type;
```

**Employee\_Track\_points\_emtype**

Emp_id	Emp_type	Avg_Disci_points
101	Normal	0
102	Normal	1
103	Full time	0
104	Supervisor	0
105	Full time	

Employee_Track_points_emptytype		
Emp_id	Emp_type	Avg_Disci_points
106	Supervisor	1
107	Normal	
108	Supervisor	
109	Supervisor	2
110	Supervisor	



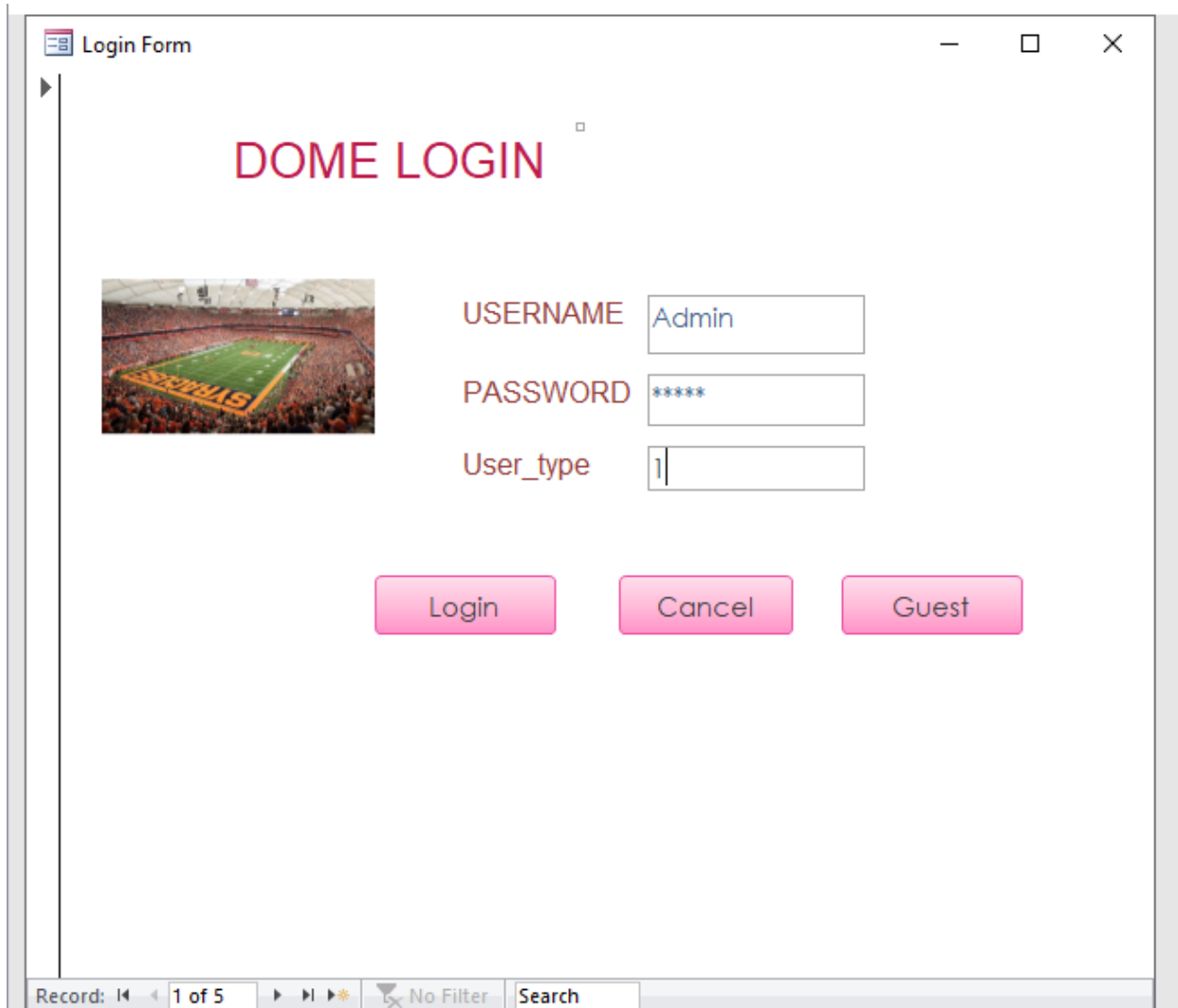
## Forms:

The users in this database application are classified as SU Students, Supervisors, Full time employees, Managers, Administrative team personnel.

Various Forms are created for the front end of the Database Application which is connected with Microsoft SQL Server in the backend.

## Login Form:


The below Master Login Form allows the SU employees and the other workers of the Dome to login and manage their schedules. The form has 1 for Admin personnel. The form once the person successfully logs in as Admin User or other User takes them to the Navigation page which is different for Admin and other users.



The screenshot shows a web application window titled "Login Form". The main heading is "DOME LOGIN" in large, bold, red letters. To the left of the login fields is a small image of a baseball field. The login fields are labeled "USERNAME", "PASSWORD", and "User\_type". The "USERNAME" field contains the text "Admin". The "PASSWORD" field contains five asterisks "\*\*\*\*\*". The "User\_type" field contains the number "1". Below the fields are three pink buttons: "Login", "Cancel", and "Guest". At the bottom of the window, there is a status bar with the text "Record: 1 of 5", a "No Filter" button, and a "Search" input field.

Welcome Navigation Page:

This form is a welcome navigation page which opens on successful login for the Admin User. Logout is created to logout of this page and go back to Login Page. The Admin Person cannot change the Manager details or Employee Details.



Welcome Page

Logout

Payroll

dbo\_Employee


Track Records

Game Schedule

EmployeeSchedule

Manager Details

Dome Concessions Payroll



Pay\_id

41

Emp\_id

101

Emp\_type

Normal

Pay\_rate

11.6

Hours\_recorded

0


Amount\_paid

0

Normal User Login:

Login Form

DOMELOGIN



USERNAME

akas2505

PASSWORD

\*\*\*\*\*

User\_type

6

Login

Cancel

Guest

Record: 1 of 5

No Filter

Search

Welcome for Normal Users which doesn't allow the users to modify contents and other details.

Navigation Form\_new

Welcome Page

Logout

EmployeeSchedule

Game Schedule

DropShift

Payroll

dbo\_Employee

AvailableShift

Employee Schedule

Emp\_id

101

Shift\_id

201

Shift\_date

9/7/2019

Shift\_type


Beer

Shift\_time

17:30:00

Stand\_number

147B



## Reports Form:

The form created has 3 buttons for each individual report created. The manager can go to this page and click on the button for the report he/she wants to see. This form makes it easier for the managers to view the reports which can be created without having to search or look for it.

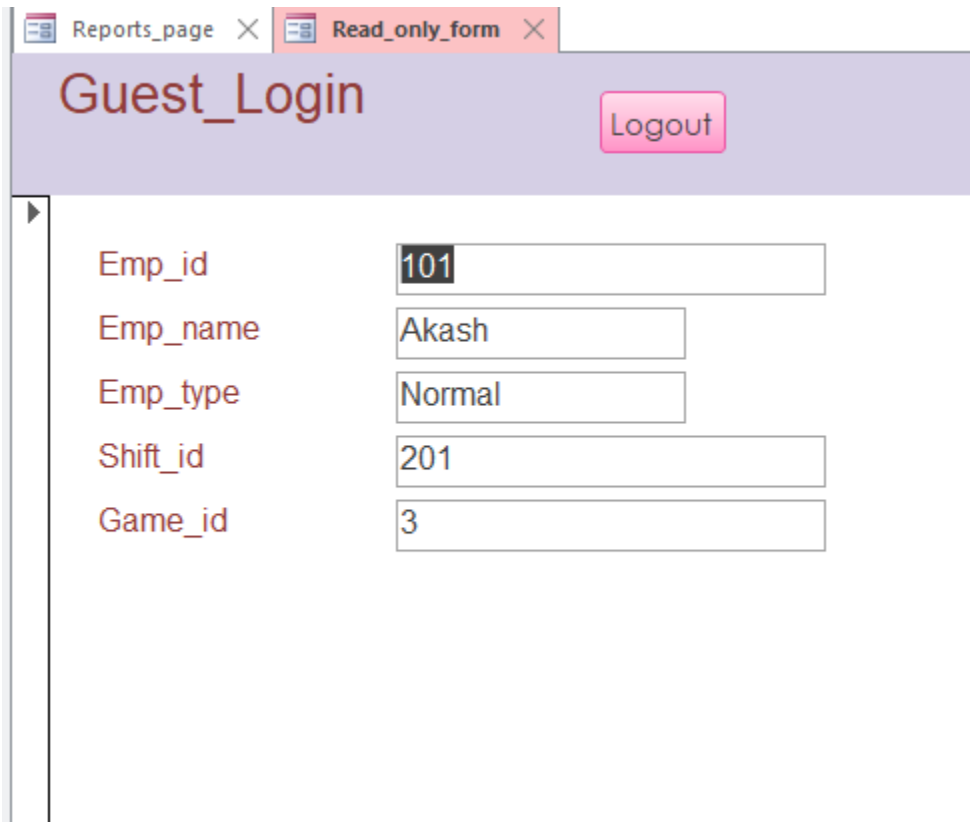
**For this form I have created Employee\_type as a “Combo Box” on a new Form.**

The screenshot shows a web browser window with a tab titled "Reports\_page". The main content area is titled "Report Page" in a large, dark red font. Below the title, there is a label "Emp\_type\_working" followed by a dropdown menu. The dropdown menu is currently set to "Full time" and has a downward arrow icon. Below the dropdown menu, there are three pink buttons with rounded corners, arranged horizontally. The buttons are labeled "Active\_Emp", "Reasons\_Drop", and "Track\_emp".

**Guest Form:**

This form is a read only form for the Guest Users which pops out on Clicking Guest Button on the Main Login Page. The details cannot be modified instead can only be viewed.

This is a sub form created by using Employee, Shift Schedule & Game tables fields taken together.



Guest_Login	
Emp_id	101
Emp_name	Akash
Emp_type	Normal
Shift_id	201
Game_id	3

### Employee Details Form:



This is a Employee Details form created to show the employee details. The fields can be changed by only Managers and not other persons.

There are 2 more buttons on this form which can be used to search a record from this form and go to last record to add a new employee detail named Add Employee.

Reports\_page X Manager Details X dbo\_Employee X

## Employee Details

Emp_id	101
Emp_name	Akash
Emp_address	118 Concord Place
Emp_contactNo	3158804123
Emp_emailID	aks123@gmail.com
Emp_type	Normal


Game Schedule Form:

The form created displays the Game Schedule for all the Games scheduled to be played at the Dome with their timings, date, teams and whether it is a Basketball or Football Game.

New Games can be added by Managers only so that it is visible to all the employees working in the Dome.

Reports\_page X Manager Details X dbo\_Employee X Game Schedule X

Game Schedule



Game\_id

1

Game\_type

football

Game\_time

19:00:00

Game\_teams

Syracuse Boston

Game\_date

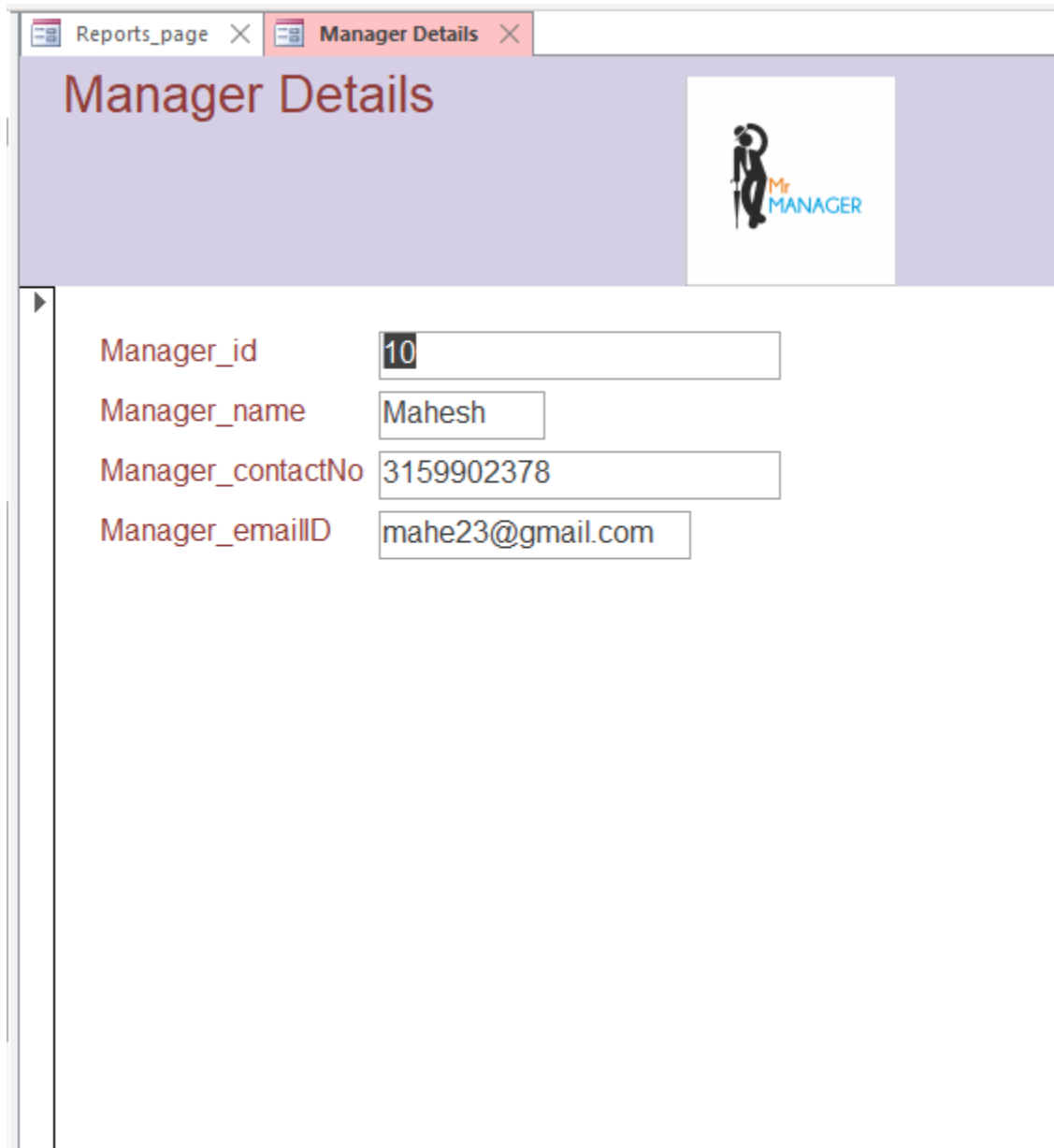
9/7/2019

Add Game



## Manager Details:

The form created here shows the details of the Manager working at the Dome. Their details cannot be edited by anyone else but only by themselves.



The screenshot shows a web browser window with two tabs: 'Reports\_page' and 'Manager Details'. The 'Manager Details' tab is active, displaying a form titled 'Manager Details' with a logo on the right. The logo features a stylized figure and the text 'Mr MANAGER'. The form contains four input fields with the following values:

Field Name	Value
Manager_id	10
Manager_name	Mahesh
Manager_contactNo	3159902378
Manager_emailID	mahe23@gmail.com

Shift Schedule Form:

The Shift Schedule Form contains all the information about the shifts for the games. The shift timings, date, type and duration for a particular game is enlisted in this form.

Reports\_page

ShiftSchedule

ShiftSchedule

Shift\_id

201

Shift\_date

9/7/2019

Shift\_time

17:30:00

Shift\_type

Beer

Shift\_duration

5

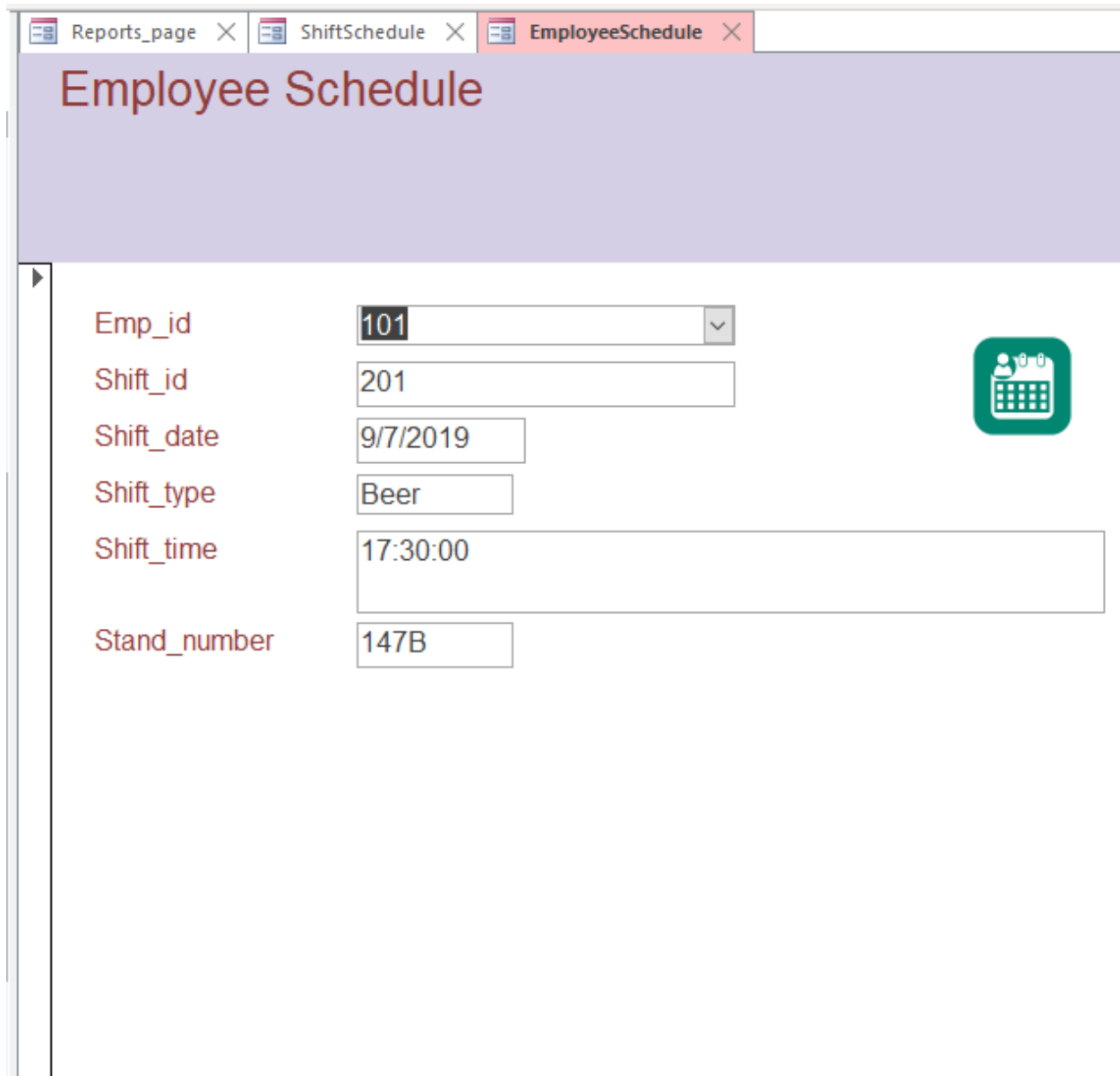
Game\_id

1

**Employee Schedule Form:**

This form helps the employees to look into what their schedule is which shift they have taken and all the other details of the shift taken by them. It allows the employees to figure out about their working hours and plan other work/study accordingly.

**Here Emp\_id has “Combo Box” which is a primary Key in Employee Table.**



Employee Schedule

Emp_id	101
Shift_id	201
Shift_date	9/7/2019
Shift_type	Beer
Shift_time	17:30:00
Stand_number	147B

Payroll Form:

The Payroll form is used to display the pay related queries. An Employee can view how many hours they have logged in while working along with the amount that would be paid to them by Dome.


It will also help employees to see their hours worked so they do not go over the limit of 20 hours and plan on taking/dropping shifts as per the hours remaining with them.

Reports\_page

Payroll

Dome Concessions

Payroll



Pay\_id

41

Emp\_id

101

Emp\_type

Normal

Pay\_rate

11.6

Hours\_recorded

0

Amount\_paid

0

Track Records Form:

The form enables the managers/supervisors to see how the employees perform while working for a game. The feedback is collected from the Supervisor and Manager after each game to keep track of the efficiency of the employees.

The disciplinary points are a measure of inconsistency or not doing their work as per the set standard. These points are given to them based on being late, lazy or other aspects which make them less efficient.

Reports\_page

Payroll

Track Records

Track Records

Track\_id

21

Emp\_id


101

Feedback

ontime good

Disci\_points

0



Available Shift Form:

This form shows the shifts available to be picked up by the employees using the Pick Button.

Reports\_page X Payroll X Track Records X AvailableShift X

AvailableShift

Avail\_id

Shift\_id

Emp\_id

31

210

101

Pick

### Drop Shift Form:

This form shows the Dropped shifts by the employees. The Employees can drop the shift by using the Drop Button in the form.

The Manager\_id is 1001 by default for all the dropped shifts as they are not yet approved, 1001 is kind of not yet approved or manager has approved it.

For approved dropped shifts it would include the Manager\_id of the Manager that approves this drop request of any employee.

The screenshot shows a web application window with multiple tabs: Reports\_page, Payroll, Track Records, AvailableShift, and DropShift. The DropShift tab is active, displaying a form titled "DropShift". The form contains five input fields with the following labels and values:

Field Label	Value
Drop_id	51
Manager_id	1001
Shift_id	204
Emp_id	101
Reason	Exceeding hours

Below the input fields is a pink button labeled "Drop".

## PROCEDURE:

This procedure was created for updating the Hours\_recorded for an employee in the Pay. The hours\_recorded is a derived attribute from the shift duration.

**Logic:** The logic behind creating this procedure was to automatically update the total number of hours based on shift duration for an employee so that he can be paid based on the hours he/she has worked.

The query calculates sum of the hours for an employee which is the sum of shift duration for all the shifts the person works which comes from his schedule in the Employee Schedule(Emp Schedule) Table

Once the procedure is run, it checks for matching shifts that all the employees in their shift schedule table have and calculates sum of the hours on shift duration for an employee.

The output for this before and after running is the same as for Trigger which is shown in the next section where this procedure is executed, and it updates the hours for each employee.

```
CREATE PROCEDURE Hours_recorded
AS
Begin
UPDATE Pay
SET Hours_recorded = Hrs.Hours_recorded
FROM (select ems.emp_id,sum(shift.Shift_duration) Hours_recorded from EmpSchedule ems INNER JOIN
ShiftSchedule shft ON shft.Shift_id=ems.Shift_id group by ems.emp_id) As Hrs
where Pay.Emp_id=Hrs.Emp_id
END;
```



## TRIGGER:

I implemented a trigger to automatically update the Amount\_Paid column value which is a derived attribute automatically as and when any changes are performed on the Pay table to record the Payable Amount.

**Logic:** The logic is that whenever the Pay table is updated the trigger running on Pay would make sure that the Amount Paid is updated automatically and no manual intervention is required.

The Amount Paid is calculated for each time the Pay table has some changes in it which is calculated as  $\text{Hours\_recorded} * \text{Pay\_rate}$ .

```
CREATE TRIGGER Amt_Paid
ON Pay
FOR INSERT,UPDATE
AS
IF @@ROWCOUNT >=1
Begin
Update Pay SET Amount_Paid = Pd.Amount_Paid
FROM (Select Emp_id,(Hours_recorded*Pay_rate) Amount_Paid from Pay) As Pd
where Pay.Emp_id=Pd.Emp_id
END;
```

### Before Running Trigger Output of Pay Table:

Select \* from Pay

100 %

Results Messages

	Pay_id	Emp_id	Emp_type	Pay_rate	Hours_recorded	Amount_paid
1	41	101	Normal	11.60	0.00	0.00
2	42	102	Normal	11.60	0.00	0.00
3	43	103	Full time	15.00	0.00	0.00
4	44	104	Supervisor	13.00	0.00	0.00
5	45	106	Supervisor	13.00	0.00	0.00
6	46	107	Normal	11.60	0.00	0.00
7	47	108	Normal	11.60	0.00	0.00
8	48	109	Supervisor	13.00	0.00	0.00
9	49	110	Supervisor	13.00	0.00	0.00
10	50	105	Full time	15.00	0.00	0.00

### Executing the Procedure to update Hours\_recorded which makes changes in the Pay table thereby invoking the Trigger to run automatically:

Procedure and Trigger run one after other affecting 10 rows in Hours\_recorded column which causes trigger to update Amount\_Paid for all the 10 records in the Pay Table.

EXEC Hours\_recorded

100 %

Messages

(10 rows affected)

(10 rows affected)

**Output of Pay Table After Trigger is run:**

Select \* from Pay

100 %

Results Messages

	Pay_id	Emp_id	Emp_type	Pay_rate	Hours_recorded	Amount_paid
1	41	101	Normal	11.60	15.00	174.00
2	42	102	Normal	11.60	10.00	116.00
3	43	103	Full time	15.00	10.00	150.00
4	44	104	Supervisor	13.00	10.00	130.00
5	45	106	Supervisor	13.00	4.00	52.00
6	46	107	Normal	11.60	5.00	58.00
7	47	108	Normal	11.60	5.00	58.00
8	48	109	Supervisor	13.00	5.00	65.00
9	49	110	Supervisor	13.00	6.00	78.00
10	50	105	Full time	15.00	5.00	75.00