

Name : Pratik Bharat Tamhankar

CWID: A20384093

## Table of Contents

PHASE V.....	2
Review of the Current Project Status: .....	2
PHASE VI.....	3
Designing the Physical Application .....	3
Additional Tables Proposed by Leadership Team.....	22
PHASE VII.....	24
PHASE VIII.....	32
Testing the Database System.....	33
PHASE IX.....	46
Data Analytics Performed .....	46
WEB Presence .....	51
PHASE X.....	55
Systems Analysis and Viewpoints .....	55

## PHASE V

### Review of the Current Project Status:

Currently, as per the project timelines, below are the key milestones which are achieved as of now:

- Creation of Data Team
- Stated Assumptions
- Conceptual Modelling
- Logical Modelling
- Analytics Operations possible

Looking the above models developed, we shall now move ahead with below steps in the project

- Creation of Physical models
- Apply the required constraints for data sanity purpose
- Perform quality sample data insertions
- Test the constraints
- Perform additions in the tables if required
- Identify the possible data analytics operations and reports the organization can achieve based on the data models created

## PHASE VI

### Designing the Physical Application

Below is the list of tables which were defined in the previous phases along with its descriptions and implementation details

Sr No	Entity Name	Table Name	DDL
1	DEPARTMENT	trvl_adv_department_pbt	create table trvl_adv_department_pbt (Dept_Id number primary key, Dept_Name varchar2(30), Dept_Loc varchar2(30) );
2	EMPLOYEES	trvl_Adv_employees_pbt	create table trvl_Adv_employees_pbt ( Emp_ID number primary key, Emp_Name varchar2(30), emp_Dept_id number, Emp_designation varchar2(30) not null, Emp_phone number, Emp_Hire_Date date, Emp_Mgr_Id number, constraint fk_dep_id foreign key (emp_Dept_id) references trvl_adv_department_pbt(Dept_id), constraint fk_mngr_id foreign key (emp_mngr_id) references trvl_Adv_employees_pbt(Emp_ID) );
3	PROJECTS	trvl_adv_proj_pbt	create table trvl_adv_proj_pbt ( Proj_ID varchar2(30) primary key, Proj_Name varchar2(255) not null, Proj_Start_Date date, Proj_End_Date date, Proj_Mgr_Id number, Proj_Cost number, constraint fk_pmgr_id foreign key (Proj_Mgr_Id) references

			trvl_Adv_employees_pbt(Emp_ID));
4	EVENTS	trvl_adv_events_pbt	<pre>create table trvl_adv_events_pbt( Event_Id varchar2(30) primary key, Proj_Id varchar2(30), Event_Desc varchar2(255), Event_loc varchar2(255) not null, Event_start_date date, Event_end_date date, Event_Mgr number, constraint fk_proj_id foreign key(Proj_Id) references trvl_adv_proj_pbt(Proj_ID) );</pre>
5	TA_REQUEST	trvl_adv_request_pbt	<pre>create table trvl_adv_request_pbt ( TA_Request_Id number primary key, Requester_ID number, Event_Id varchar2(30), Proj_ID varchar2(30), TA_Remittance_Type varchar2(30) not null, Trvl strt_date date, Trvl end_date date, TA_Apporval_EMP_ID number, constraint fk_emp_id2 foreign key (TA_Apporval_EMP_ID) references trvl_Adv_employees_pbt(Emp_ID), constraint fk_evnt_id1 foreign key(event_id) references trvl_adv_events_pbt(Event_Id), constraint fk_proj_id2 foreign key(Proj_Id) references trvl_adv_proj_pbt(Proj_ID) );</pre>
6	TA_RESPONSE	trvl_adv_response_pbt	<pre>create table trvl_adv_response_pbt ( TA_RESP_ID number primary key, Event_Id varchar2(30), Emp_ID number, Project_Id varchar2(30), Total_Expenses number, Response_Date date, constraint fk_evnt_id3 foreign key(event_id) references</pre>

			trvl_adv_events_pbt(Event_Id), constraint fk_emp_id3 foreign key (Emp_ID) references trvl_Adv_employees_pbt(Emp_ID), constraint fk_proj_id3 foreign key(Project_Id) references trvl_adv_proj_pbt(Proj_ID) );
7	TRAVEL_ADVANCES	trvl_adv_travel_Advances_pbt	create table trvl_adv_travel_Advances_pbt( TA_Issuance_Id number primary key, TA_Request_Id number, Event_Id varchar2(30), TA strt_date date, TA_end_date date, TA_Remittance_Type varchar2(30), TA_Amount number, TA_Status varchar2(30), Dept_Id number, constraint fk_dept_id foreign key(dept_Id) references trvl_adv_department_pbt(dept_id), constraint fk_evnt_id2 foreign key(event_id) references trvl_adv_events_pbt(Event_Id), constraint ta_req_id foreign key(TA_Request_Id) references trvl_adv_request_pbt(ta_request_id) );
8	EXPENSES	trvl_adv_exp_details_pbt	create table trvl_adv_exp_details_pbt (Exp_Id varchar2(30) primary key, Event_Id varchar2(30), Proj_Id varchar2(30), Emp_ID number, Exp_type varchar2(255), Exp_Cost number, Exp_desc varchar2(255), constraint fk_evnt_id foreign key(event_id) references trvl_adv_events_pbt(Event_Id), constraint fk_proj_id1 foreign key(Proj_Id) references trvl_adv_proj_pbt(Proj_ID), constraint fk_emp_id1 foreign key (Emp_ID) references trvl_Adv_employees_pbt(Emp_ID)

			) ;
9	SETTLEMENT_DETAILS	trvl_adv_Settle_details_pbt	create table trvl_adv_Settle_details_pbt ( Set_ID number, Set_Event_ID varchar2(30), Set_issuance_ID number, Set_Utilization_Type varchar2(30), Set_total_expenses number, Set_total_advances number, Set_Initiated_Date date, Set_last_modified_Date date, constraint fk_evnt_id4 foreign key(Set_Event_ID) references trvl_adv_events_pbt(Event_Id), constraint fk_iss_id foreign key(Set_issuance_ID) references trvl_adv_request_pbt(ta_request_id) );

Table Creation Screenshots:

### 1: DEPARTMENT TABLE

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there are two expandable code snippets. The first snippet contains the SQL code for creating the 'trvl\_adv\_department\_pbt' table. The second snippet contains the SQL code for creating the 'trvl\_Adv\_employees\_pbt' table, which includes a foreign key reference to the first table. A yellow callout box in the top-right corner displays the student's name and ID: 'Name: Pratik Tamhankar' and 'Student Id: A20384093'. At the bottom of the interface, the 'Script Output' tab is active, showing the message 'Task completed in 0.039 seconds' and the confirmation 'table TRVL\_ADV\_DEPARTMENT\_PBT created.'

```
create table trvl_adv_department_pbt
(Dept_Id number primary key,
Dept_Name varchar2(30),
Dept_Loc varchar2(30)
);

create table trvl_Adv_employees_pbt
(
Emp_ID number primary key,
Emp_Name varchar2(30) DEFAULT 'Unknown Name',
Emp_dept_id number foreign key references trvl_adv_department_pbt(Dept_Id),
Emp_designation varchar2(30) not null,
Emp_phone number,
```

Script Output | Task completed in 0.039 seconds  
table TRVL\_ADV\_DEPARTMENT\_PBT created.

### 2: EMPLOYEES Table

The screenshot shows the Oracle SQL Developer interface. It features a single expandable code snippet in the top-left pane containing the SQL code for creating the 'trvl\_Adv\_employees\_pbt' table. This table includes fields like Emp\_ID, Emp\_Name, emp\_Dept\_id, Emp\_designation, Emp\_phone, Emp\_Hire\_Date, and Emp\_Mgr\_Id, along with two foreign key constraints referencing the first table. A yellow callout box in the top-right corner displays the student's name and ID: 'Name: Pratik Tamhankar' and 'Student Id: A20384093'. The 'Script Output' tab at the bottom is active, showing the message 'Task completed in 0.049 seconds' and the confirmation 'table TRVL\_ADV\_EMPLOYEES\_PBT created.'

```
create table trvl_Adv_employees_pbt
(
Emp_ID number primary key,
Emp_Name varchar2(30),
emp_Dept_id number,
Emp_designation varchar2(30) not null,
Emp_phone number,
Emp_Hire_Date date,
Emp_Mgr_Id number ,
constraint fk_dep_id foreign key (emp_Dept_id) references trvl_adv_department_pbt(Dept_id),
constraint fk_mgr_id foreign key (emp_mngr_id) references trvl_Adv_employees_pbt(Emp_ID)
);
```

Script Output | Task completed in 0.049 seconds  
table TRVL\_ADV\_EMPLOYEES\_PBT created.

## 3: PROJECT table

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL script to create the `TRVL_ADV_PROJ_PBT` table. The table has columns for `Proj_ID`, `Proj_Name`, `Proj_Start_Date`, `Proj_End_Date`, `Proj_Mgr_Id`, and `Proj_Cost`. It includes a foreign key constraint `fk_pmgr_id` referencing the `Emp_ID` column in the `trvl_Adv_employees_pbt` table. Below the code editor is a "Script Output" window showing the message "Task completed in 0.044 seconds". A yellow callout box in the bottom-right corner displays the student information: Name: Pratik Tamhankar and Student Id: A20384093.

```
create table trvl_adv_proj_pbt(
    Proj_ID varchar2(30) primary key,
    Proj_Name varchar2(255) not null,
    Proj_Start_Date date,
    Proj_End_Date date,
    Proj_Mgr_Id number,
    Proj_Cost number,
    constraint fk_pmgr_id foreign key (Proj_Mgr_Id) references trvl_Adv_employees_pbt(Emp_ID));
```

table TRVL\_ADV\_PROJ\_PBT created.

Name: Pratik Tamhankar  
Student Id: A20384093

## 4: EVENTS Table

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL script to create the `TRVL_ADV_EVENTS_PBT` table. The table has columns for `Event_Id`, `Proj_Id`, `Event_Desc`, `Event_loc`, `Event_start_date`, `Event_end_date`, and `Event_Mgr`. It includes a foreign key constraint `fk_proj_id` referencing the `Proj_ID` column in the `trvl_adv_proj_pbt` table. Below the code editor is a "Script Output" window showing the message "Task completed in 0.05 seconds". A yellow callout box in the bottom-right corner displays the student information: Name: Pratik Tamhankar and Student Id: A20384093.

```
create table trvl_adv_events_pbt(
    Event_Id varchar2(30) primary key,
    Proj_Id varchar2(30),
    Event_Desc varchar2(255),
    Event_loc varchar2(255) not null,
    Event_start_date date,
    Event_end_date date,
    Event_Mgr number,
    constraint fk_proj_id foreign key(Proj_Id) references trvl_adv_proj_pbt(Proj_ID)
);
```

table TRVL\_ADV\_EVENTS\_PBT created.

Name: Pratik Tamhankar  
Student Id: A20384093

5: EXPENSES table

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL script to create the table:

```
create table trvl_adv_exp_details_pbt
(Exp_Id varchar2(30) primary key,
Event_Id varchar2(30),
Proj_Id varchar2(30),
Emp_ID number,
Exp_type varchar2(255),
Exp_Cost number,
Exp_desc varchar2(255),
constraint fk_evnt_id foreign key(event_id) references trvl_adv_events_pbt(Event_Id),
constraint fk_proj_idl foreign key(Proj_Id) references trvl_adv_proj_pbt(Proj_ID),
constraint fk_emp_idl foreign key (Emp_ID) references trvl_Adv_employees_pbt(Emp_ID)
);
```

In the bottom-left pane, there is a "Script Output" window showing the result of the execution:

table TRVL\_ADV\_EXP\_DETAILS\_PBT created.

A yellow callout box is overlaid on the output window, containing the following information:

+ ... Name: Pratik Tamhankar  
Student Id: A20384093

## 6: REQUEST Table

```
create table trvl_adv_request_pbt
(
    TA_Request_Id number primary key,
    Requester_ID number,
    Event_Id varchar2(30),
    Proj_ID varchar2(30),
    TA_Remittance_Type varchar2(30) not null,
    Trvl strt_date date,
    Trvl_end_date date,
    TA_Apporval_EMP_ID number,
    constraint fk_emp_id2 foreign key (TA_Apporval_EMP_ID) references trvl_Adv_employees_pbt(Emp_ID),
    constraint fk_evnt_idl foreign key(event_id) references trvl_adv_events_pbt(Event_Id),
    constraint fk_proj_id2 foreign key(Proj_Id) references trvl_adv_proj_pbt(Proj_ID)
);
```

Script Output x | Task completed in 0.071 seconds

table TRVL\_ADV\_REQUEST\_PBT created.

+

...

✖

Name: Pratik Tamhankar

Student Id: A20384093

## 7: TRAVEL ADVANCE Table

```
create table trvl_adv_travel_Advances_pbt(
    TA_Issuance_Id number primary key,
    TA_Request_Id number,
    Event_Id varchar2(30),
    TA strt_date date,
    TA_end_date date,
    TA_Remittance_Type varchar2(30),
    TA_Amount number,
    TA_Status varchar2(30),
    Dept_Id number,
    constraint fk_dept_id foreign key(dept_Id) references trvl_adv_department_pbt(dept_id),
    constraint fk_evnt_id2 foreign key(event_id) references trvl_adv_events_pbt(Event_Id),
    constraint ta_req_id foreign key(TA_Request_Id) references trvl_adv_request_pbt(ta_request_id)
);
```

Script Output x  
Task completed in 0.049 seconds

table TRVL\_ADV\_TRAVEL\_ADVANCES\_PBT created.

+

Name: Pratik Tamhankar  
Student Id: A20384093

## 8: RESPONSE Table

```
create table trvl_adv_response_pbt(
    TA_RESP_ID number primary key,
    Event_Id varchar2(30),
    Emp_ID number,
    Project_Id varchar2(30),
    Total_Expenses number,
    Response_Date date,
    constraint fk_evnt_id3 foreign key(event_id) references trvl_adv_events_pbt(Event_Id),
    constraint fk_emp_id3 foreign key (Emp_ID) references trvl_Adv_employees_pbt(Emp_ID),
    constraint fk_proj_id3 foreign key(Project_Id) references trvl_adv_proj_pht(Proj_ID)
);
```

+

Name: Pratik Tamhankar

Student Id: A20384093

Script Output | Task completed in 0.052 seconds

table TRVL\_ADV\_RESPONSE\_PBT created.

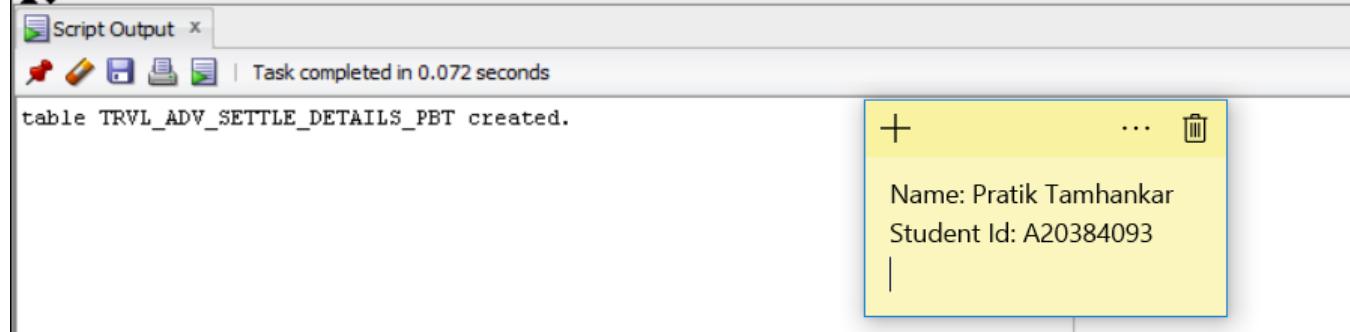
## 9: SETTLEMENT DETAILS Table

;

```

create table trvl_adv_Settle_details_pbt
(
Set_ID number,
Set_Event_ID varchar2(30),
Set_Issuance_ID number,
Set_Utilization_Type varchar2(30),
Set_Total_Expenses number,
Set_Total_Advances number,
Set_Initiated_Date date,
Set_Last_Modified_Date date,
constraint fk_evt_id4 foreign key(Set_Event_ID) references trvl_adv_events_pbt(Event_Id),
constraint fk_iss_id foreign key(Set_Issuance_ID) references trvl_adv_request_pbt(ta_request_id)
);

```



Modifications Performed to the created Table

SQL Queries:

```

alter table trvl_adv_request_pbt add requested_amount number;
alter table trvl_adv_response_pbt add approved_Amount number;
alter table trvl_adv_response_pbt add approval_status varchar2(20);
alter table trvl_adv_events_pbt add event_status varchar2(20);
alter table trvl_adv_response_pbt rename column EMP_ID to responded_by_emp;
alter table trvl_adv_response_pbt add ta_req_id number;
alter table trvl_adv_travel_Advances_pbt add foreign key(TA_RESPONSE_ID) references
trvl_adv_response_pbt(ta_resp_id);
alter table trvl_adv_travel_Advances_pbt add TA_RESPONSE_ID number;
alter table trvl_adv_travel_Advances_pbt add TA_RESPONSE_ID number;

```

Sequences created:

```

CREATE SEQUENCE trvl_req_sequence START WITH 4000 INCREMENT BY 1 NOCACHE NOCYCLE;
CREATE SEQUENCE trvl_res_sequence START WITH 8000 INCREMENT BY 1 NOCACHE NOCYCLE;
CREATE SEQUENCE trvl_TA_sequence START WITH 66666 INCREMENT BY 1 NOCACHE NOCYCLE;
drop sequence trvl_TA_sequence;
CREATE SEQUENCE trvl_SETTLE START WITH 919191 INCREMENT BY 1 NOCACHE NOCYCLE;

```

Insertions Performed:

### 1: DEPARTMENT Table

	DEPT_ID	DEPT_NAME	DEPT_LOC
1	1	SALES	NEW YORK
2	2	PROJECT MANAGEMENT	NEW YORK
3	3	ACCOUNTS	CHICAGO
4	4	IT	CHICAGO
5	5	OPERATIONS	SEATTLE
6	6	MIS	BOSTON
7	7	MARKETING	NEW YORK
8	8	RESEARCH AND DEVELOPMENT	SAN FRANCISCO
9	9	HUMAN RESOURCE	MIAMI
10	10	ENGINEERING	LA

+ ...

Name: Pratik Tamhankar  
Student Id: A20384093

### 2: EMPLOYEES Table

	EMP_ID	EMP_NAME	EMP_DEPT_ID	EMP_DESIGNATION	EMP_PHONE	EMP_HIRE_DATE	EMP_MGR_ID
1	3001 ARJUN	2 PROJECT MANAGER HEAD	3121111813	01-JUN-15	3001		
2	3002 YASH	2 PROJECT MANAGER	3121111823	01-JUN-16	3001		
3	3003 ARJUN	2 PROJECT MANAGER	3121111834	01-JUL-16	3001		
4	9001 LILLY	3 ACCOUNTING HEAD	7127111889	13-AUG-16	9001		
5	9002 RIA	3 ACCOUNTANT	3127879099	12-MAY-16	9001		
6	9003 RIA	3 ACCOUNTANT	3127879022	11-MAR-16	9001		
7	7001 STATHAM	1 SALES HEAD	5123568867	08-JUL-15	7001		
8	7002 JASON	1 SALES REPRESENTATIVE	5123568876	18-SEP-16	7001		
9	7003 JESSICA	1 SALES REPRESENTATIVE	5123568855	11-AUG-16	7001		
10	7004 JOHN	1 SALES REPRESENTATIVE	5123568850	19-SEP-16	7001		

+ ...

Name: Pratik Tamhankar  
Student Id: A20384093

## 3: PROJECT Table

Screenshot of Oracle SQL Developer showing the TRVL\_ADV\_PROJ\_PBT table.

Table Structure:

```

    +-----+-----+-----+-----+-----+
    | PROJ_ID | PROJ_NAME        | PROJ_START_DATE | PROJ_END_DATE | PROJ_MGR_ID | PROJ_COST |
    +-----+-----+-----+-----+-----+
    | 1 1001 | Accounting reconciliation framework | 01-JAN-16      | 31-MAR-16    | 3001       | 75000    |
    | 2 1002 | Project Loon          | 31-MAR-16      | 30-APR-16    | 3002       | 28000    |
    | 3 1003 | Waymo                  | 30-APR-16      | 30-MAY-16    | 3003       | 75000    |
    | 4 1004 | OpenRefine             | 15-JAN-16      | 14-FEB-16    | 3001       | 39000    |
    | 5 1005 | Google Brain           | 25-FEB-16      | 26-MAR-16    | 3002       | 75000    |
    | 6 1006 | Project Soli            | 31-MAR-16      | 30-APR-16    | 3003       | 65000    |
    | 7 1007 | Project Ara             | 15-APR-16      | 15-MAY-16    | 3001       | 75000    |
    | 8 1008 | Robo Army               | 01-DEC-16      | 31-DEC-16    | 3002       | 43000    |
    | 9 1009 | Project Tango            | 15-JAN-17      | 14-FEB-17    | 3003       | 75000    |
    | 10 1010 | Social Media Presence   | 25-FEB-17      | 27-MAR-17    | 3001       | 39000    |
    +-----+-----+-----+-----+-----+
  
```

Yellow callout box:

Name: Pratik Tamhankar  
Student Id: A20384093

## 4: EVENTS Table

Screenshot of Oracle SQL Developer showing the TRVL\_ADV\_EVENTS\_PBT table.

Table Structure:

```

    +-----+-----+-----+-----+-----+-----+-----+-----+
    | EVENT_ID | PROJ_ID | EVENT_DESC        | EVENT_LOC     | EVENT_START_DATE | EVENT_END_DATE | EVENT_MGR |
    +-----+-----+-----+-----+-----+-----+-----+
    | 1 E101  | 1001   | Pre-Sales Meet    | CHICAGO      | 01-JAN-16       | 07-JAN-16     | 101      |
    | 2 E102  | 1002   | Sales Bid Meet    | NEWYORK      | 08-JAN-16       | 20-JAN-16     | 102      |
    | 3 E103  | 1003   | Pre-Sales Meet    | DALLAS       | 21-JAN-16       | 28-JAN-16     | 103      |
    | 4 E104  | 1001   | Customer Meet     | WISCONSIN    | 04-FEB-16       | 11-FEB-16     | 104      |
    | 5 E105  | 1002   | Contract bid meet | BOSTON       | 18-FEB-16       | 25-FEB-16     | 105      |
    | 6 E106  | 1003   | Salers meet       | VEGAS        | 26-FEB-16       | 04-MAR-16     | 106      |
    | 7 E107  | 1001   | Infotech produc... | MIAMI        | 05-MAR-16       | 12-MAR-16     | 107      |
    | 8 E108  | 1007   | Sales meetup      | CHICAGO     | 13-MAR-16       | 20-MAR-16     | 108      |
    | 9 E109  | 1008   | Salers meet       | NEWYORK     | 21-MAR-16       | 28-MAR-16     | 109      |
    | 10 E110 | 1009   | Pre-Sales Meet    | CHICAGO     | 29-MAR-16       | 05-APR-16     | 110      |
    +-----+-----+-----+-----+-----+-----+-----+
  
```

Yellow callout box:

Name: Pratik Tamhankar  
Student Id: A20384093

## 5:Expenses:

Columns Data Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

EXP_ID	EVENT_ID	PROJ_ID	EMP_ID	EXP_TYPE	EXP_COST	EXP_DESC
1 1	E110	1007	7002	Air Travel	1897	To and Fro travel till event location and back
2 11	E109	1008	7003	Meals	1925	Meals during the event, before the event and after the event
3 12	E108	1009	7002	Business Travel Insurance	5500	Special Insurance as per the location was required
4 13	E102	1007	7003	Event Registration fees	750	NA
5 14	E106	1008	7002	Training cost	1100	Additional training cost wasn't included in Original Event Itinerary
6 2	E101	1001	7003	Travel	400	To and Fro travel till airport, event location and back
7 3	E101	1001	7002	Telephone	350	Calls to the vendors overseas
8 4	E103	1003	7003	Meals	200	Meals before the event date weren't included
9 5	E104	1001	7002	Business Travel Insurance	3000	Special Insurance as per the location was required
10 6	E105	1002	7003	Event registration fees	500	Was changed at the last moment
11 7	E105	1002	7004	Meals	390	Meals before the event date weren't included
12 10	E107	1001	7004	Meals	1150	Meals before the event date weren't included
13 8	E101	1001	7003	Lodging	450	Re-accommodation was done due to safety issues
14 9	E103	1003	7003	Lodging	700	Re-accommodation was done due to safety issues

## 6: REQUESTS

Query:

```

insert into trvl_adv_request_pbt
select trvl_req_sequence.nextval,exp.emp_id,exp.event_id,exp.proj_id,exp.remittance_type,
(select event_start_Date from trvl_adv_events_pbt event where event.event_id=exp.event_id)
event_start_Date,
(select event_end_Date from trvl_adv_events_pbt event where event.event_id=exp.event_id) event_end_Date,
(select emp.emp_mgr_id from trvl_Adv_employees_pbt emp where emp.emp_id=exp.emp_id) approved_by,
exp.requested_Amount
from (
select
exp.emp_id,exp.event_id,exp.proj_id,'TRAVEL CARD' remittance_type
,sum(exp.exp_cost) requested_Amount
from trvl_adv_exp_details_pbt exp
group by
exp.emp_id,exp.event_id,exp.proj_id,'TRAVEL CARD'

```

) exp;

TA_REQUEST_ID	REQUESTER_ID	EVENT_ID	PROJ_ID	TA_REMITTANCE_TYPE	TRLV_STRT_DATE	TRLV_END_DATE	TA_APPORVAL_EMP_ID	REQUESTED_AMOUNT
1	4000	7002 E106	1008	CASH	26-FEB-16	04-MAR-16	7001	1100
2	4001	7004 E107	1001	TRAVEL CARD	05-MAR-16	12-MAR-16	7001	1150
3	4002	7002 E110	1007	CASH	29-MAR-16	05-APR-16	7001	1897
4	4003	7003 E103	1003	TRAVEL CARD	21-JAN-16	28-JAN-16	7003	900
5	4004	7004 E105	1002	TRAVEL CARD	18-FEB-16	25-FEB-16	7001	390
6	4005	7002 E104	1001	CASH	04-FEB-16	11-FEB-16	7001	3000
7	4006	7002 E108	1009	TRAVEL CARD	13-MAR-16	20-MAR-16	7002	5500
8	4007	7003 E102	1007	TRAVEL CARD	08-JAN-16	20-JAN-16	7001	750
9	4008	7003 E105	1002	CASH	18-FEB-16	25-FEB-16	7001	500
10	4009	7003 E109	1008	TRAVEL CARD	21-MAR-16	28-MAR-16	7003	1925
11	4010	7003 E101	1001	CASH	01-JAN-16	07-JAN-16	7001	850
12	4011	7002 E101	1001	TRAVEL CARD	01-JAN-16	07-JAN-16	7001	350

+

...

Name: Pratik Tamhankar  
Student Id: A20384093

7: RESPONSES:

**Query:**

```
insert into trvl_adv_response_pbt
select trvl_res_sequence.nextval,response.event_id,response.responded_by_Emp,response.proj_id,
response.approval_Date,response.ta_request_id,response.requested_amount,
decode(
(select emp_Designation from trvl_Adv_employees_pbt
where emp_id=response.ta_apporval_emp_id),'SALES HEAD','APPROVED','REJECTED')
approval_stat
from (
select req.event_id, (select round(dbms_random.value(9001,9003)) num from dual)
responded_by_Emp,req.proj_id,
sysdate approval_Date,
req.requested_amount,
req.requester_id,
req.ta_request_id,
req.ta_apporval_emp_id
from
trvl_adv_request_pbt req
) response;
```

**Output:**

The screenshot shows the Oracle SQL Developer interface with the 'Data' tab selected for the 'TRVL\_ADV\_RESPONSE\_PBT' table. The table contains 12 rows of data, each representing a response record with columns: TA\_RESP\_ID, EVENT\_ID, RESPONDED\_BY\_EMP, PROJECT\_ID, RESPONSE\_DATE, TA\_REQ\_ID, TA\_REQ\_AMT, and APPROVAL\_STATUS.

	TA_RESP_ID	EVENT_ID	RESPONDED_BY_EMP	PROJECT_ID	RESPONSE_DATE	TA_REQ_ID	TA_REQ_AMT	APPROVAL_STATUS
1	8000	E106		9003 1008	08-MAY-17	4000	1100	APPROVED
2	8001	E107		9003 1001	08-MAY-17	4001	1150	APPROVED
3	8002	E110		9003 1007	08-MAY-17	4002	1897	APPROVED
4	8003	E103		9003 1003	08-MAY-17	4003	900	REJECTED
5	8004	E105		9003 1002	08-MAY-17	4004	390	APPROVED
6	8005	E104		9003 1001	08-MAY-17	4005	3000	APPROVED
7	8006	E108		9003 1009	08-MAY-17	4006	5500	REJECTED
8	8007	E102		9003 1007	08-MAY-17	4007	750	APPROVED
9	8008	E105		9003 1002	08-MAY-17	4008	500	APPROVED
10	8009	E109		9003 1008	08-MAY-17	4009	1925	REJECTED
11	8010	E101		9003 1001	08-MAY-17	4010	850	APPROVED
12	8011	E101		9003 1001	08-MAY-17	4011	350	APPROVED

A yellow callout box in the bottom right corner displays the student's name and ID:

Name: Pratik Tamhankar  
Student Id: A20384093

## 8: TRAVEL ADVANCES

**Query:**

```
insert into trvl_adv_travel_Advances_pbt
select trvl_TA_sequence.nextval,resp.event_id,resp.response_Date,resp.response_Date+30,
(
select ta_remittance_type from trvl_adv_request_pbt req
where req.ta_request_id=resp.ta_Req_id
) approval_stat,ta_req_amt,'OPEN',
(
select emp_dept_id from trvl_Adv_employees_pbt where emp_id=(Select requester_id from
trvl_adv_request_pbt where ta_request_id=resp.ta_Req_id
)
) as "DEPT",resp.ta_resp_id
from trvl_adv_response_pbt resp
where resp.approval_status='APPROVED';
```

**Output:**

The screenshot shows the Oracle SQL Developer interface with a query results window. The query has inserted 9 rows into the TRVL\_ADV\_TRAVEL\_ADVANCES\_PBT table. A tooltip at the bottom right of the results pane displays the user information: Name: Pratik Tamhankar and Student Id: A20384093.

TA_ISSUANCE_ID	EVENT_ID	TA_STRT_DATE	TA_END_DATE	TA_REMITTANCE_TYPE	TA_AMOUNT	TA_STATUS	DEPT_ID	TA_RESPONSE_ID
1	66687 E106	08-MAY-17	07-JUN-17	CASH	1100	OPEN	1	8000
2	66688 E107	08-MAY-17	07-JUN-17	TRAVEL CARD	1150	OPEN	1	8001
3	66689 E110	08-MAY-17	07-JUN-17	CASH	1897	OPEN	1	8002
4	66690 E105	08-MAY-17	07-JUN-17	TRAVEL CARD	390	OPEN	1	8004
5	66691 E104	08-MAY-17	07-JUN-17	CASH	3000	OPEN	1	8005
6	66692 E102	08-MAY-17	07-JUN-17	TRAVEL CARD	750	OPEN	1	8007
7	66693 E105	08-MAY-17	07-JUN-17	CASH	500	OPEN	1	8008
8	66694 E101	08-MAY-17	07-JUN-17	CASH	850	OPEN	1	8010
9	66695 E101	08-MAY-17	07-JUN-17	TRAVEL CARD	350	OPEN	1	8011

## 9: SETTLEMENT

**Query:**

```
insert into trvl_adv_settle_details_pbt --
(set_id, set_event_id, set_issuance_id, set_total_expenses, set_total_advances, set_initiated_date, set_modified_
date)
select trvl_SETTLE.nextval, settle_abc.* from
(
select event_id, ta_req_id issuance_id, "util_Type",
(select sum(exp_cost) expenses from trvl_adv_exp_details_pbt exp where resp.event_id=exp.event_id and
resp.project_id=exp.proj_id) expenses,
(select sum(requested_Amount) advances from trvl_adv_request_pbt req where
req.ta_request_id=resp.ta_req_id) advances,
sysdate set_initiated_date,
sysdate set_last_modified_date
from trvl_adv_response_pbt resp
) settle_abc;
commit;
```

**Trigger on Settlement table:**

```
create or replace
trigger trig_settlement_after_insert
AFTER INSERT
    ON trvl_adv_settle_details_pbt
```

```
DECLARE
v_expenses number;
v_request number;
v_util_type varchar2(30);

BEGIN

for i in (
select * from trvl_adv_settle_details_pbt
) loop
v_Expenses:=i.SET_TOTAL_EXPENSES;
v_request:=i.SET_TOTAL_ADVANCES;

if(nvl(v_Expenses,0)>nvl(v_request,0)) then
v_util_type:='OVER UTILIZED';
else if (nvl(v_Expenses,0)<nvl(v_request,0)) then
v_util_type:='UNDER UTILIZED';
else
v_util_type:='BALANCED';
```

```
end if;  
end if;
```

```
update trvl_adv_settle_details_pbt u set u.SET_UTILIZATION_TYPE=v_util_type where u.SET_ID=i.set_id;  
dbms_output.put_line('in trigger');  
end loop;  
END;
```

**Output:**

The screenshot shows a database interface with a table named `TRL_ADV_SETTLE_DETAILS_PBT`. The table has columns: `SET_ID`, `SET_EVENT_ID`, `SET_ISSUANCE_ID`, `SET_UTILIZATION_TYPE`, `SET_TOTAL_EXPENSES`, `SET_TOTAL_ADVANCES`, `SET_INITIATED_DATE`, and `SET_LAST_MODIFIED_DATE`. The data consists of 12 rows, each containing a unique `SET_ID` and `SET_EVENT_ID` pair, along with their respective utilization types, total expenses, advances, and dates.

	SET_ID	SET_EVENT_ID	SET_ISSUANCE_ID	SET_UTILIZATION_TYPE	SET_TOTAL_EXPENSES	SET_TOTAL_ADVANCES	SET_INITIATED_DATE	SET_LAST_MODIFIED_DATE
1	919203	E101		4011 OVER UTILIZED	1200	350	08-MAY-17	08-MAY-17
2	919204	E101		4010 OVER UTILIZED	1200	850	08-MAY-17	08-MAY-17
3	919205	E102		4007 BALANCED	750	750	08-MAY-17	08-MAY-17
4	919206	E103		4003 BALANCED	900	900	08-MAY-17	08-MAY-17
5	919207	E104		4005 BALANCED	3000	3000	08-MAY-17	08-MAY-17
6	919208	E105		4008 OVER UTILIZED	890	500	08-MAY-17	08-MAY-17
7	919209	E105		4004 OVER UTILIZED	890	390	08-MAY-17	08-MAY-17
8	919210	E106		4000 BALANCED	1100	1100	08-MAY-17	08-MAY-17
9	919211	E107		4001 BALANCED	1150	1150	08-MAY-17	08-MAY-17
10	919212	E108		4006 BALANCED	5500	5500	08-MAY-17	08-MAY-17
11	919213	E109		4009 BALANCED	1925	1925	08-MAY-17	08-MAY-17
12	919214	E110		4002 BALANCED	1897	1897	08-MAY-17	08-MAY-17

+

...

Name: Pratik Tamhankar  
Student Id: A20384093

## Additional Tables Proposed by Leadership Team

As per the recommendations of the leadership team, below is an additional table for getting the historical data of the employees, their travel advance requests and total expenses recorded till date.

### Create Table Query:

```
CREATE TABLE "ORA_PTAMHANKAR"."TRVL_ADV_OPERATIONAL_DATA"
(
    "EMP_NAME" VARCHAR2(30 BYTE),
    "EMP_DESIGNATION" VARCHAR2(30 BYTE) NOT NULL ENABLE,
    "EMP_PHONE" NUMBER,
    "DEPT_NAME" VARCHAR2(30 BYTE),
    "EXP_DESC" VARCHAR2(255 BYTE),
    "EXP_COST" NUMBER,
    "EXP_TYPE" VARCHAR2(255 BYTE),
    "REQUESTED_AMOUNT" NUMBER,
    "TA_REMITTANCE_TYPE" VARCHAR2(30 BYTE) NOT NULL ENABLE,
    "TA_REQ_AMT" NUMBER,
    "APPROVAL_STATUS" VARCHAR2(20 BYTE)
);
```

### Query:

```
SELECT
D.EMP_NAME,
D.EMP_DESIGNATION,
D.EMP_PHONE,
C.DEPT_NAME,
E.EXP_DESC,
E.EXP_COST,
E.EXP_TYPE,
F.REQUESTED_AMOUNT,
F.TA_REMITTANCE_TYPE,
G.TA_REQ_AMT,
G.APPROVAL_STATUS
FROM TRVL_ADV_EMPLOYEES_PBT D,
TRVL_ADV_DEPARTMENT_PBT C,
TRVL_ADV_EXP_DETAILS_PBT E,
TRVL_ADV_REQUEST_PBT F,
TRVL_ADV_RESPONSE_PBT G
WHERE C.DEPT_ID = D.EMP_DEPT_ID
AND D.EMP_ID = E.EMP_ID
AND D.EMP_ID = F.TA_APPORVAL_EMP_ID
AND F.TA_REQUEST_ID = G.TA_REQ_ID;
```

**Output:**

The screenshot shows a database query results grid from Oracle SQL Developer. The query is run against the TRVL\_ADV\_OPERATIONAL\_DATA table. The results show various travel expenses for employees Jason and Jessica. A yellow callout box at the bottom left contains the student's information.

EMP_NAME	EMP_DESIGNATION	EMP_PHONE	DEPT_NAME	EXP_DESC	EXP_COST	EXP_TYPE	REQUESTED_AMOUNT	TA_REMI...	TA_REQ_AMT	APPROVAL
1 JASON	SALES REPRESENTATIVE	5123568876	SALES	To and Fro travel till ...	1897	Air Travel	5500	TRAVEL CARD	5500	REJECTED
2 JESSICA	SALES REPRESENTATIVE	5123568855	SALES	Meals during the event,...	1925	Meals	1925	TRAVEL CARD	1925	REJECTED
3 JESSICA	SALES REPRESENTATIVE	5123568855	SALES	Meals during the event,...	1925	Meals	900	TRAVEL CARD	900	REJECTED
4 JASON	SALES REPRESENTATIVE	5123568876	SALES	Special Insurance as pe...	5500	Business Travel ...	5500	TRAVEL CARD	5500	REJECTED
5 JESSICA	SALES REPRESENTATIVE	5123568855	SALES	NA	750	Event Registrati...	1925	TRAVEL CARD	1925	REJECTED
6 JESSICA	SALES REPRESENTATIVE	5123568855	SALES	NA	750	Event Registrati...	900	TRAVEL CARD	900	REJECTED
7 JASON	SALES REPRESENTATIVE	5123568876	SALES	Additional training cos...	1100	Training cost	5500	TRAVEL CARD	5500	REJECTED
8 JESSICA	SALES REPRESENTATIVE	5123568855	SALES	Lunch Details	150	Lunch	1925	TRAVEL CARD	1925	REJECTED
9 JESSICA	SALES REPRESENTATIVE	5123568855	SALES	Lunch Details	150	Lunch	900	TRAVEL CARD	900	REJECTED
10 JESSICA	SALES REPRESENTATIVE	5123568855	SALES	To and Fro travel till ...	400	Travel	1925	TRAVEL CARD	1925	REJECTED
11 JESSICA	SALES REPRESENTATIVE	5123568855	SALES	To and Fro travel till ...	400	Travel	900	TRAVEL CARD	900	REJECTED
12 JASON	SALES REPRESENTATIVE	5123568876	SALES	Calls to the vendors ov...	350	Telephone	5500	TRAVEL CARD	5500	REJECTED
13 JESSICA	SALES REPRESENTATIVE	5123568855	SALES	Meals before the event ...	200	Meals	1925	TRAVEL CARD	1925	REJECTED
14 JESSICA	SALES REPRESENTATIVE	5123568855	SALES	Meals before the event ...	200	Meals	900	TRAVEL CARD	900	REJECTED
15 JASON	SALES REPRESENTATIVE	5123568876	SALES	Special Insurance as pe...	3000	Business Travel ...	5500	TRAVEL CARD	5500	REJECTED
16 JESSICA	SALES REPRESENTATIVE	5123568855	SALES	Was changed at the last...	500	Event registrati...	1925	TRAVEL CARD	1925	REJECTED
17 JESSICA	SALES REPRESENTATIVE	5123568855	SALES	Was changed at the last...	500	Event registrati...	900	TRAVEL CARD	900	REJECTED
18 JESSICA	SALES REPRESENTATIVE	5123568855	SALES	Re-accomodation was don...	450	Lodging	1925	TRAVEL CARD	1925	REJECTED
19 JESSICA	SALES REPRESENTATIVE	5123568855	SALES	Re-accomodation was don...	450	Lodging	900	TRAVEL CARD	900	REJECTED
20 JESSICA	SALES REPRESENTATIVE	5123568855	SALES	Re-accomodation was don...	700	Lodging	1925	TRAVEL CARD	1925	REJECTED
21 JESSICA	SALES REPRESENTATIVE	5123568855	SALES	Re-accomodation was don...	700	Lodging	900	TRAVEL CARD	900	REJECTED

+ ... Name: Pratik Tamhankar  
Student Id: A20384093

## PHASE VII

Loading the Tables with Valid Data

Table 1:

The screenshot shows the Oracle SQL Developer interface with the 'DEPARTMENT\_PBT' table selected. The table has three columns: DEPT\_ID, DEPT\_NAME, and DEPT\_LOC. The data is as follows:

DEPT_ID	DEPT_NAME	DEPT_LOC
1	SALES	NEW YORK
2	PROJECT MANAGEMENT	NEW YORK
3	ACCOUNTS	CHICAGO
4	IT	CHICAGO
5	OPERATIONS	SEATTLE
6	MIS	BOSTON
7	MARKETING	NEW YORK
8	RESEARCH AND DEVELOPMENT	SAN FRANCISCO
9	HUMAN RESOURCE	MIAMI
10	ENGINEERING	LA

A yellow callout box in the bottom right corner contains the following text:  
Name: Pratik Tamhankar  
Student Id: A20384093

Table 2:

The screenshot shows a database interface with a toolbar at the top and a table named 'EMPLOYEES' below it. The table has columns: EMP\_ID, EMP\_NAME, EMP\_DEPT\_ID, EMP\_DESIGNATION, EMP\_PHONE, EMP\_HIRE\_DATE, and EMP\_MGR\_ID. The data is as follows:

	EMP_ID	EMP_NAME	EMP_DEPT_ID	EMP_DESIGNATION	EMP_PHONE	EMP_HIRE_DATE	EMP_MGR_ID
1	3001	ARJUN	2	PROJECT MANAGER HEAD	3121111813	01-JUN-15	3001
2	3002	YASH	2	PROJECT MANAGER	3121111823	01-JUN-16	3001
3	3003	ZEUS	2	PROJECT MANAGER	3121111834	01-JUL-16	3001
4	9001	LILLY	3	ACCOUNTING HEAD	7127111889	13-AUG-16	9001
5	9002	RIA	3	ACCOUNTANT	3127879099	12-MAY-16	9001
6	9003	RIA	3	ACCOUNTANT	3127879022	11-MAR-16	9001
7	7001	STATHAM	1	SALES HEAD	5123568867	08-JUL-15	7001
8	7002	JASON	1	SALES REPRESENTATIVE	5123568876	18-SEP-16	7001
9	7003	JESSICA	1	SALES REPRESENTATIVE	5123568855	11-AUG-16	7001
10	7004	JOHN	1	SALES REPRESENTATIVE	5123568850	19-SEP-16	7001

A yellow callout box highlights the last row (John) with the following information:  
Name: Pratik Tamhankar  
Student Id: A20384093

Table 3:

The screenshot shows a database interface with several tabs at the top: Proj\_1.sql (selected), Data\_Insert.sql, and TRL\_ADV\_EVENTS\_PBT. Below the tabs, there are buttons for Columns, Data, Constraints, Grants, Statistics, Triggers, Flashback, Dependencies, Details, Partitions, Indexes, and SQL. A toolbar with icons for new, edit, delete, and search follows. The main area displays a table with the following data:

	EVENT_ID	PROJ_ID	EVENT_DESC	EVENT_LOC	EVENT_START_DATE	EVENT_END_DATE	EVENT_MGR	EVENT_STATUS
1	E101	1001	Pre-Sales Meet	CHICAGO	01-JAN-16	07-JAN-16	101	OPEN
2	E102	1002	Sales Bid Meet	NEWYORK	08-JAN-16	20-JAN-16	102	CANCELED
3	E103	1003	Pre-Sales Meet	DALLAS	21-JAN-16	28-JAN-16	103	OPEN
4	E104	1001	Customer Meet	WISCONSIN	04-FEB-16	11-FEB-16	104	CLOSED
5	E105	1002	Contract bid meet	BOSTON	18-FEB-16	25-FEB-16	105	OPEN
6	E106	1003	Salers meet	VEGAS	26-FEB-16	04-MAR-16	106	CANCELED
7	E107	1001	Infotech product line launch	MIAMI	05-MAR-16	12-MAR-16	107	OPEN
8	E108	1007	Sales meetup	CHICAGO	13-MAR-16	20-MAR-16	108	CLOSED
9	E109	1008	Salers meet	NEWYORK	21-MAR-16	28-MAR-16	109	OPEN
10	E110	1009	Pre-Sales Meet	CHICAGO	29-MAR-16	05-APR-16	110	CLOSED

A yellow callout box in the bottom-left corner contains the following text:  
Name: Pratik Tamhankar  
Student Id: A20384093

Table 4:

Columns Data Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

Sort. Filter:

	EXP_ID	EVENT_ID	PROJ_ID	EMP_ID	EXP_TYPE	EXP_COST	EXP_DESC
1	1	E110	1007	7002	Air Travel	1897	To and Fro travel till event location and back
2	11	E109	1008	7003	Meals	1925	Meals during the event, before the event and after the event
3	12	E108	1009	7002	Business Travel Insurance	5500	Special Insurance as per the location was required
4	13	E102	1007	7003	Event Registration fees	750	NA
5	14	E106	1008	7002	Training cost	1100	Additional training cost wasn't included in Original Event Itinerary
6	2	E101	1001	7003	Travel	400	To and Fro travel till airport, event location and back
7	3	E101	1001	7002	Telephone	350	Calls to the vendors overseas
8	4	E103	1003	7003	Meals	200	Meals before the event date weren't included
9	5	E104	1001	7002	Business Travel Insurance	3000	Special Insurance as per the location was required
10	6	E105	1002	7003	Event registration fees	500	Was changed at the last moment
11	7	E105	1002	7004	Meals	390	Meals before the event date weren't included
12	10	E107	1001	7004	Meals	1150	Meals before the event date weren't included
13	8	E101	1001	7003	Lodging	450	Re-accommodation was done due to safety issues
14	9	E103	1003	7003	Lodging	700	Re-accommodation was done due to safety issues

+    ...    ✖  
 Name: Pratik Tamhankar  
 Student Id: A20384093

Table 5:

The screenshot shows a database interface with three tabs at the top: Proj\_1.sql, Data\_Insert.sql, and TRL\_ADV\_PROJ\_PBT. The TRL\_ADV\_PROJ\_PBT tab is active, displaying a table with the following schema and data:

	PROJ_ID	PROJ_NAME	PROJ_START_DATE	PROJ_END_DATE	PROJ_MGR_ID	PROJ_COST
1	1001	Accounting reconciliation framework	01-JAN-16	31-MAR-16	3001	75000
2	1002	Project Loon	31-MAR-16	30-APR-16	3002	28000
3	1003	Waymo	30-APR-16	30-MAY-16	3003	75000
4	1004	OpenRefine	15-JAN-16	14-FEB-16	3001	39000
5	1005	Google Brain	25-FEB-16	26-MAR-16	3002	75000
6	1006	Project Soli	31-MAR-16	30-APR-16	3003	65000
7	1007	Project Ara	15-APR-16	15-MAY-16	3001	75000
8	1008	Robo Army	01-DEC-16	31-DEC-16	3002	43000
9	1009	Project Tango	15-JAN-17	14-FEB-17	3003	75000
10	1010	Social Media Presence	25-FEB-17	27-MAR-17	3001	39000

A yellow callout box is positioned over the bottom right corner of the table area, containing the following text:

+ ... Name: Pratik Tamhankar  
Student Id: A20384093

Table 6:

The screenshot shows the Oracle SQL Developer interface with the 'Data' tab selected for the 'TRVL\_ADV\_REQUEST\_PBT' table. The table contains 12 rows of data, each representing a travel request. A yellow callout box highlights the last row (ID 4011) with the following details:

TA_REQUEST_ID	REQUESTER_ID	EVENT_ID	PROJ_ID	TA_REMITTANCE_TYPE	TRVL_STRT_DATE	TRVL_END_DATE	TA_APPORVAL_EMP_ID	REQUESTED_AMOUNT
1	4000	7002 E106	1008	CASH	26-FEB-16	04-MAR-16	7001	1100
2	4001	7004 E107	1001	TRAVEL CARD	05-MAR-16	12-MAR-16	7001	1150
3	4002	7002 E110	1007	CASH	29-MAR-16	05-APR-16	7001	1897
4	4003	7003 E103	1003	TRAVEL CARD	21-JAN-16	28-JAN-16	7003	900
5	4004	7004 E105	1002	TRAVEL CARD	18-FEB-16	25-FEB-16	7001	390
6	4005	7002 E104	1001	CASH	04-FEB-16	11-FEB-16	7001	3000
7	4006	7002 E108	1009	TRAVEL CARD	13-MAR-16	20-MAR-16	7002	5500
8	4007	7003 E102	1007	TRAVEL CARD	08-JAN-16	20-JAN-16	7001	750
9	4008	7003 E105	1002	CASH	18-FEB-16	25-FEB-16	7001	500
10	4009	7003 E109	1008	TRAVEL CARD	21-MAR-16	28-MAR-16	7003	1925
11	4010	7003 E101	1001	CASH	01-JAN-16	07-JAN-16	7001	850
12	4011	7002 E101	1001	TRAVEL CARD	01-JAN-16	07-JAN-16	7001	350

Callout box details:  
Name: Pratik Tamhankar  
Student Id: A20384093

Table 7:

The screenshot shows a database interface with several tabs at the top: Proj\_1.sql, Data\_Insert.sql, and TRL\_ADV\_RESPONSE\_PBT (which is the active tab). Below the tabs are navigation links: Columns, Data, Constraints, Grants, Statistics, Triggers, Flashback, Dependencies, Details, Partitions, Indexes, and SQL. A toolbar with icons for refresh, insert, delete, and search follows. The main area displays a table with the following data:

	TA_RESP_ID	EVENT_ID	RESPONDED_BY_EMP	PROJECT_ID	RESPONSE_DATE	TA_REQ_ID	TA_REQ_AMT	APPROVAL_STATUS
1	8000	E106		9003 1008	08-MAY-17	4000	1100	APPROVED
2	8001	E107		9003 1001	08-MAY-17	4001	1150	APPROVED
3	8002	E110		9003 1007	08-MAY-17	4002	1897	APPROVED
4	8003	E103		9003 1003	08-MAY-17	4003	900	REJECTED
5	8004	E105		9003 1002	08-MAY-17	4004	390	APPROVED
6	8005	E104		9003 1001	08-MAY-17	4005	3000	APPROVED
7	8006	E108		9003 1009	08-MAY-17	4006	5500	REJECTED
8	8007	E102		9003 1007	08-MAY-17	4007	750	APPROVED
9	8008	E105		9003 1002	08-MAY-17	4008	500	APPROVED
10	8009	E109		9003 1008	08-MAY-17	4009	1925	REJECTED
11	8010	E101		9003 1001	08-MAY-17	4010	850	APPROVED
12	8011	E101		9003 1001	08-MAY-17	4011	350	APPROVED

A yellow callout box is positioned over the last row (row 12) of the table, containing the following information:

+ ...

Name: Pratik Tamhankar  
Student Id: A20384093

Table 8:

The screenshot shows a database interface with several tabs at the top: Proj\_1.sql, Data\_Insert.sql, and TRVL\_ADV\_SETTLE\_DETAILS\_PBT (which is the active tab). Below the tabs is a toolbar with icons for refresh, undo, redo, and other operations. The main area displays a table with the following columns: SET\_ID, SET\_EVENT\_ID, SET\_ISSUANCE\_ID, SET\_UTILIZATION\_TYPE, SET\_TOTAL\_EXPENSES, SET\_TOTAL\_ADVANCES, SET\_INITIATED\_DATE, and SET\_LAST\_MODIFIED\_DATE. The data consists of 12 rows, each containing a unique ID and event ID, along with utilization type, total expenses, total advances, and dates. A yellow callout box highlights the last row (row 12), which corresponds to the student information provided in the question.

	SET_ID	SET_EVENT_ID	SET_ISSUANCE_ID	SET_UTILIZATION_TYPE	SET_TOTAL_EXPENSES	SET_TOTAL_ADVANCES	SET_INITIATED_DATE	SET_LAST_MODIFIED_DATE
1	919203	E101		4011 OVER UTILIZED	1200	350	08-MAY-17	08-MAY-17
2	919204	E101		4010 OVER UTILIZED	1200	850	08-MAY-17	08-MAY-17
3	919205	E102		4007 BALANCED	750	750	08-MAY-17	08-MAY-17
4	919206	E103		4003 BALANCED	900	900	08-MAY-17	08-MAY-17
5	919207	E104		4005 BALANCED	3000	3000	08-MAY-17	08-MAY-17
6	919208	E105		4008 OVER UTILIZED	890	500	08-MAY-17	08-MAY-17
7	919209	E105		4004 OVER UTILIZED	890	390	08-MAY-17	08-MAY-17
8	919210	E106		4000 BALANCED	1100	1100	08-MAY-17	08-MAY-17
9	919211	E107		4001 BALANCED	1150	1150	08-MAY-17	08-MAY-17
10	919212	E108		4006 BALANCED	5500	5500	08-MAY-17	08-MAY-17
11	919213	E109		4009 BALANCED	1925	1925	08-MAY-17	08-MAY-17
12	919214	E110		4002 BALANCED	1897	1897	08-MAY-17	08-MAY-17

+ ... ⚡

Name: Pratik Tamhankar

Student Id: A20384093

Table 9:

The screenshot shows a database interface with several tabs at the top: Proj\_1.sql, Data\_Insert.sql, and TRL\_ADV\_TRAVEL\_ADVANCES\_PBT (which is the active tab). Below the tabs are navigation links: Columns, Data, Constraints, Grants, Statistics, Triggers, Flashback, Dependencies, Details, Partitions, Indexes, and SQL. Underneath these are toolbar icons for insert, update, delete, and search, followed by Sort.. and Filter.. buttons.

The main area displays a table with the following columns and data:

	TA_ISSUANCE_ID	EVENT_ID	TA_STRT_DATE	TA_END_DATE	TA_REMITTANCE_TYPE	TA_AMOUNT	TA_STATUS	DEPT_ID	TA_RESPONSE_ID
1	66687	E106	08-MAY-17	07-JUN-17	CASH	1100	OPEN	1	8000
2	66688	E107	08-MAY-17	07-JUN-17	TRAVEL CARD	1150	OPEN	1	8001
3	66689	E110	08-MAY-17	07-JUN-17	CASH	1897	OPEN	1	8002
4	66690	E105	08-MAY-17	07-JUN-17	TRAVEL CARD	390	OPEN	1	8004
5	66691	E104	08-MAY-17	07-JUN-17	CASH	3000	OPEN	1	8005
6	66692	E102	08-MAY-17	07-JUN-17	TRAVEL CARD	750	OPEN	1	8007
7	66693	E105	08-MAY-17	07-JUN-17	CASH	500	OPEN	1	8008
8	66694	E101	08-MAY-17	07-JUN-17	CASH	850	OPEN	1	8010
9	66695	E101	08-MAY-17	07-JUN-17	TRAVEL CARD	350	OPEN	1	8011

A yellow callout box in the bottom right corner contains the following text:  
Name: Pratik Tamhankar  
Student Id: A20384093

## PHASE VIII

## Testing the Database System

Checking of constraints:

1: DEPARTMENT table:

**Integrity constraint on Primary Key testing:**

1.1 Trapping Invalid Data:

The screenshot shows an Oracle SQL Developer interface. In the script editor at the top, there is a single line of SQL code: `insert into trvl_adv_Department_pbt (dept_id,dept_name,dept_loc) values (10,'LOGISTICS','OHIO');`. Below the editor, a toolbar has icons for Query Result, Script Output, and others. A status bar indicates "Task completed in 0.323 seconds". The main pane displays an error message starting at line 182: "Error starting at line 182 in command: insert into trvl\_adv\_Department\_pbt (dept\_id,dept\_name,dept\_loc) values (10,'LOGISTICS','OHIO')". It continues with an "Error report" section containing the following details:

- SQL Error: ORA-00001: unique constraint (ORA\_PTAMHANKAR.SYS\_C0064865) violated 00001. 00000 - "unique constraint (%s.%s) violated"
- \*Cause: An UPDATE or INSERT statement attempted to insert a duplicate key. For Trusted Oracle configured in DBMS MAC mode, you may see this message if a duplicate entry exists at a different level.
- \*Action: Either remove the unique restriction or do not insert the key.

A yellow callout box on the right side contains the text "Name: Pratik Tamhankar" and "Student Id: A20384093".

1.2 Passing Valid Data:

The screenshot shows an Oracle SQL Developer interface. In the script editor at the top, there is a single line of SQL code: `insert into trvl_adv_Department_pbt (dept_id,dept_name,dept_loc) values (11,'LOGISTICS','OHIO');`. Below the editor, a toolbar has icons for Query Result, Script Output, and others. A status bar indicates "Task completed in 0.088 seconds". The main pane displays the message "1 rows inserted." A yellow callout box on the right side contains the text "Name: Pratik Tamhankar" and "Student Id: A20384093".

1.3 Trapping Invalid Delete, as child entries exist:

```
delete from trvl_adv_Department_pbt where  
dept_id=2;
```

Query Result X | Query Result 1 X | Script Output X | Query Result 3 X | Query Result 4 X

Task completed in 0.341 seconds

Error starting at line 184 in command:

```
delete from trvl_adv_Department_pbt where  
dept_id=2
```

Error report:

```
SQL Error: ORA-02292: integrity constraint (ORA_PTAMHANKAR.FK_DEP_ID) violated - child record found  
02292. 00000 - "integrity constraint (%s.%s) violated - child record found"  
*Cause: attempted to delete a parent key value that had a foreign  
dependency.  
*Action: delete dependencies first then parent or disable constraint.
```

+ ...

Name: Pratik Tamhankar  
Student Id: A20384093

2: EMPLOYEE Table

2.1 Passing Valid Data and NULL as Foreign key Value:

The screenshot shows the Oracle SQL Developer interface. In the top query editor, a script is written to insert a new employee record into the 'trvl\_adv\_employees\_pbt' table. The script includes values for EMP\_ID, EMP\_NAME, EMP\_DEPT\_ID, EMP\_DESIGNATION, EMP\_PHONE, EMP\_HIRE\_DATE, and EMP\_MGR\_ID. The EMP\_MGR\_ID value is set to 'null'. A 'commit;' statement follows the insert. Below the editor, a message indicates '1 rows inserted.' and 'committed.'. To the right, a yellow tooltip displays the inserted data: 'Name: Pratik Tamhankar' and 'Student Id: A20384093'. At the bottom, a status bar shows 'Task completed in 0.085 seconds'.

```
insert into trvl_adv_employees_pbt(EMP_ID,EMP_NAME,EMP_DEPT_ID,EMP_DESIGNATION,EMP_PHONE,EMP_HIRE_DATE,EMP_MGR_ID)
values (9999,'TEST',5,'TEST',1234567890,'08-JAN-2017',null);
commit;
```

Query Result X | Query Result 1 X | Script Output X | Query Result 3 X | Query Result 4 X

Task completed in 0.085 seconds

1 rows inserted.  
committed.

Name: Pratik Tamhankar  
Student Id: A20384093

2.2 FOREIGN KEY CONSTRAINT check, by passing invalid EMP\_MGR\_ID

```
insert into trvl_adv_employees_pbt(EMP_ID,EMP_NAME,EMP_DEPT_ID,EMP_DESIGNATION,EMP_PHONE,EMP_HIRE_DATE,EMP_MGR_ID)
values (9999,'TEST',5,'TEST',1234567890,'08-JAN-2017',0);
```

Query Result x | Query Result 1 x | Script Output x | Query Result 3 x | Query Result 4 x

| Task completed in 0.338 seconds

Error starting at line 194 in command:

```
insert into trvl_adv_employees_pbt(EMP_ID,EMP_NAME,EMP_DEPT_ID,EMP_DESIGNATION,EMP_PHONE,EMP_HIRE_DATE,EMP_MGR_ID)
values (9999,'TEST',5,'TEST',1234567890,'08-JAN-2017',0)
```

Error report:

```
SQL Error: ORA-02291: integrity constraint (ORA_PTAMHANKAR.FK_MGR_ID) violated - parent key not found
02291. 00000 - "integrity constraint (%s.%s) violated - parent key not found"
*Cause:    A foreign key value has no matching primary key value.
*Action:   Delete the foreign key or add a matching primary key.
```

+

...

Delete

Name: Pratik Tamhankar

Student Id: A20384093

## 2.3 NOT NULL CONSTRAINT check

```
insert into trvl_adv_employees_pbt(EMP_ID,EMP_NAME,EMP_DEPT_ID,EMP_DESIGNATION,EMP_PHONE,EMP_HIRE_DATE,EMP_MGR_ID)
values (9999,'TEST',5,null,1234567890,'08-JAN-2017',null);
...
```

Query Result x | Query Result 1 x | Script Output x | Query Result 3 x | Query Result 4 x

| Task completed in 0.347 seconds

Error starting at line 194 in command:

```
insert into trvl_adv_employees_pbt(EMP_ID,EMP_NAME,EMP_DEPT_ID,EMP_DESIGNATION,EMP_PHONE,EMP_HIRE_DATE,EMP_MGR_ID)
values (9999,'TEST',5,null,1234567890,'08-JAN-2017',null)
```

Error report:

```
SQL Error: ORA-01400: cannot insert NULL into ("ORA_PTAMHANKAR"."TRVL_ADV_EMPLOYEES_PBT"."EMP_DESIGNATION")
01400. 00000 -  "cannot insert NULL into (%s)"
```

\*Cause:  
\*Action:

+

...

Name: Pratik Tamhankar  
Student Id: A20384093

2.4 Passing a non existent DEPARTMENT ID value(foreign key column)

```
insert into trvl_adv_employees_pbt(EMP_ID,EMP_NAME,EMP_DEPT_ID,EMP_DESIGNATION,EMP_PHONE,EMP_HIRE_DATE,EMP_MGR_ID)
values (9999,'TEST',55,'TEST',1234567890,'08-JAN-2017',null);
commit;
```

Query Result x | Query Result 1 x | Script Output x | Query Result 3 x | Query Result 4 x

Task completed in 0.268 seconds

Error starting at line 194 in command:

```
insert into trvl_adv_employees_pbt(EMP_ID,EMP_NAME,EMP_DEPT_ID,EMP_DESIGNATION,EMP_PHONE,EMP_HIRE_DATE,EMP_MGR_ID)
values (9999,'TEST',55,'TEST',1234567890,'08-JAN-2017',null)
```

Error report:

SQL Error: ORA-00001: unique constraint (ORA\_PTAMHANKAR.SYS\_C0064867) violated  
00001. 00000 - "unique constraint (%s.%s) violated"

\*Cause: An UPDATE or INSERT statement attempted to insert a duplicate key.  
For Trusted Oracle configured in DBMS MAC mode, you may see  
this message if a duplicate entry exists at a different level.

\*Action: Either remove the unique restriction or do not insert the key.

+ ... ✖  
Name: Pratik Tamhankar  
Student Id: A20384093

## 2.5 Primary Key Violation Check

```
insert into trvl_adv_employees_pbt(EMP_ID,EMP_NAME,EMP_DEPT_ID,EMP_DESIGNATION,EMP_PHONE,EMP_HIRE_DATE,EMP_MGR_ID)
values (7001,'TEST',1,'TEST',1234567890,'08-JAN-2017',7001);
```

Query Result x | Query Result 1 x | Script Output x | Query Result 3 x | Query Result 4 x

Task completed in 0.131 seconds

Error starting at line 194 in command:

```
insert into trvl_adv_employees_pbt(EMP_ID,EMP_NAME,EMP_DEPT_ID,EMP_DESIGNATION,EMP_PHONE,EMP_HIRE_DATE,EMP_MGR_ID)
values (7001,'TEST',1,'TEST',1234567890,'08-JAN-2017',7001)
```

Error report:

SQL Error: ORA-00001: unique constraint (ORA\_PTAMHANKAR.SYS\_C0064867) violated  
00001. 00000 - "unique constraint (%s.%s) violated"

\*Cause: An UPDATE or INSERT statement attempted to insert a duplicate key.  
For Trusted Oracle configured in DBMS MAC mode, you may see  
this message if a duplicate entry exists at a different level.

\*Action: Either remove the unique restriction or do not insert the key.

+ ... ✖  
Name: Pratik Tamhankar  
Student Id: A20384093

## 3: EVENT

### 3.1 Creating a new EVENT

Proj\_1.sql | Data\_Insert.sql | TRL\_ADV\_EVENTS\_PBT

Columns Data Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

Sort.. Filter: Actions...

EVENT_ID	PROJ_ID	EVENT_DESC	EVENT_LOC	EVENT_START_DATE	EVENT_END_DATE	EVENT_MGR	EVENT_STATUS
1 E11	1010	TEDX Sponsoring	CHICAGO	06-FEB-17	07-FEB-17		110 OPEN
2 E101	1001	Pre-Sales Meet	CHICAGO	01-JAN-16	07-JAN-16		101 OPEN
3 E102	1002	Sales Bid Meet	NEWYORK	08-JAN-16	20-JAN-16		102 CANCELED
4 E103	1003	Pre-Sales Meet	DALLAS	21-JAN-16	28-JAN-16		103 OPEN
5 E104	1001	Customer Meet	WISCONSIN	04-FEB-16	11-FEB-16		104 CLOSED
6 E105	1002	Contract bid meet	BOSTON	18-FEB-16	25-FEB-16		105 OPEN
7 E106	1003	Salers meet	VEGAS	26-FEB-16	04-MAR-16		106 CANCELED
8 E107	1001	Infotech product line launch	MIAMI	05-MAR-16	12-MAR-16		107 OPEN
9 E108	1007	Sales meetup	CHICAGO	13-MAR-16	20-MAR-16		108 CLOSED
10 E109	1008	Salers meet	NEWYORK	21-MAR-16	28-MAR-16		109 OPEN
11 E110	1009	Pre-Sales Meet	CHICAGO	29-MAR-16	05-APR-16		110 CLOSED

Row 11: ORA-01400: cannot insert NULL into ("ORA\_PTAMHANKAR"."TRLV\_ADV\_EMPLOYEES\_PBT"."EMP\_DESIGNATION")  
ORA-06512: at line 1

INSERT INTO "ORA\_PTAMHANKAR"."TRLV\_ADV\_EMPLOYEES\_PBT" (EMP\_ID, EMP\_NAME, EMP\_DEPT\_ID, EMP\_PHONE, EMP\_HIRE\_DATE, EMP\_MGR\_ID) VALUES ('9999', 'TEST', '90', '9999999999', TO\_DATE('2016-01-01', 'YYYY-MM-DD'))  
ORA-01400: cannot insert NULL into ("ORA\_PTAMHANKAR"."TRLV\_ADV\_EMPLOYEES\_PBT"."EMP\_DESIGNATION")  
ORA-06512: at line 1

One error saving changes to table "ORA\_PTAMHANKAR"."TRLV\_ADV\_EMPLOYEES\_PBT".  
Row 11: ORA-01400: cannot insert NULL into ("ORA\_PTAMHANKAR"."TRLV\_ADV\_EMPLOYEES\_PBT"."EMP\_DESIGNATION")  
ORA-06512: at line 1

Local changes cleared

Data\_Insert.sql\*: Mon May 08 19:30:50 CDT 2017

INSERT INTO "ORA\_PTAMHANKAR"."TRLV\_ADV\_EVENTS\_PBT" (EVENT\_ID, PROJ\_ID, EVENT\_DESC, EVENT\_LOC, EVENT\_START\_DATE, EVENT\_END\_DATE, EVENT\_MGR, EVENT\_STATUS) VALUES ('E11', '1010', 'TEDX Sponsoring', 'CHICAGO', '2017-02-06', '2017-02-07', null, '110 OPEN')

Commit Successful

### 3.2 Primary Key Violation

```
select * from trvl_Adv_events_pbt;
insert into trvl_Adv_events_pbt(event_id,proj_id,event_Desc,event_loc,event_start_Date,event_end_Date,event_mgr,event_status)
values('E110',1009 , 'TEST MEET' , 'NYC',sysdate,sysdate+2,111,'OPEN')
```

Query Result x | Query Result 1 x | Script Output x | Query Result 3 x | Query Result 4 x

Task completed in 0.1seconds

\*Cause: An UPDATE or INSERT statement attempted to insert a duplicate key.  
For Trusted Oracle configured in DBMS MAC mode, you may see  
this message if a duplicate entry exists at a different level.

\*Action: Either remove the unique restriction or do not insert the key.

Error starting at line 201 in command:

```
insert into trvl_Adv_events_pbt(event_id,proj_id,event_Desc,event_loc,event_start_Date,event_end_Date,event_mgr,event_status)
values('E110',1009 , 'TEST MEET' , 'NYC',sysdate,sysdate+2,111,'OPEN')
```

Error report:

SQL Error: ORA-00001: unique constraint (ORA\_PTAMHANKAR.SYS\_C0064878) violated  
00001. 00000 - "unique constraint (%s.%s) violated"

\*Cause: An UPDATE or INSERT statement attempted to insert a duplicate key.  
For Trusted Oracle configured in DBMS MAC mode, you may see  
this message if a duplicate entry exists at a different level.

\*Action: Either remove the unique restriction or do not insert the key.

### 3.3 NOT NULL CHECK for event description

```
insert into trvl_Adv_events_pbt(event_id,proj_id,event_Desc,event_loc,event_start_Date,event_end_Date,event_mgr,event_status)
values('E110',1009 ,null , 'NYC',sysdate,sysdate+2,111,'OPEN')
```

Query Result x | Query Result 1 x | Script Output x | Query Result 3 x | Query Result 4 x

Task completed in 0.117 seconds

Error starting at line 201 in command:

```
insert into trvl_Adv_events_pbt(event_id,proj_id,event_Desc,event_loc,event_start_Date,event_end_Date,event_mgr,event_status)
values('E110',1009 ,null , 'NYC',sysdate,sysdate+2,111,'OPEN')
```

Error report:

SQL Error: ORA-00001: unique constraint (ORA\_PTAMHANKAR.SYS\_C0064878) violated  
00001. 00000 - "unique constraint (%s.%s) violated"

\*Cause: An UPDATE or INSERT statement attempted to insert a duplicate key.  
For Trusted Oracle configured in DBMS MAC mode, you may see  
this message if a duplicate entry exists at a different level.

\*Action: Either remove the unique restriction or do not insert the key.

### 3.4 Foreign Key Check

```
insert into trvl_Adv_events_pbt(event_id,proj_id,event_Desc,event_loc,event_start_Date,event_end_Date,event_mgr,event_status)
values('E110',11111 , 'TEST', 'NYC',sysdate,sysdate+2,111,'OPEN')
```

Query Result X | Query Result 1 X | Script Output X | Query Result 3 X | Query Result 4 X

| Task completed in 0.485 seconds

Error starting at line 201 in command:

```
insert into trvl_Adv_events_pbt(event_id,proj_id,event_Desc,event_loc,event_start_Date,event_end_Date,event_mgr,event_status)
values('E110',11111 , 'TEST', 'NYC',sysdate,sysdate+2,111,'OPEN')
```

Error report:

```
SQL Error: ORA-00001: unique constraint (ORA_PTAMHANKAR.SYS_C0064878) violated
00001. 00000 - "unique constraint (%s.%s) violated"
*Cause: An UPDATE or INSERT statement attempted to insert a duplicate key.
        For Trusted Oracle configured in DBMS MAC mode, you may see
        this message if a duplicate entry exists at a different level.
*Action: Either remove the unique restriction or do not insert the key.
```

+ ... ⌂

Name: Pratik Tamhankar  
Student Id: A20384093

4: EXPENSE table

## 4.1 Adding an Expense entry in the table

The screenshot shows an Oracle SQL Developer interface. In the central query editor, the following SQL code is executed:

```
insert into trvl_adv_exp_Details_pbt (exp_id,event_id,proj_id,emp_id,exp_type,exp_cost,exp_Desc)
values
(15, 'E110',1007,7003,'Lunch',150,'Lunch Details')
commit;
```

The output window below the editor shows:

Query Result x | Query Result 1 x | Script Output x | Query Result  
Task completed in 0.1 seconds  
1 rows inserted.  
committed.

A yellow callout box on the right side of the interface displays the user's information:

Name: Pratik Tamhankar  
Student Id: A20384093

## 4.2 Primary Key Check by inserting duplicate entry in the EXPENSE table

The screenshot shows an Oracle SQL Developer interface. In the central query editor, the same SQL code as in the previous section is executed:

```
insert into trvl_adv_exp_Details_pbt (exp_id,event_id,proj_id,emp_id,exp_type,exp_cost,exp_Desc)
values
(15, 'E110',1007,7003,'Lunch',150,'Lunch Details')
commit;
```

The output window shows the result of the insertion:

Query Result x | Query Result 1 x | Script Output x | Query Result 3 x | Query Result 4 x | Query Result 5 x  
Task completed in 0.301 seconds  
1 rows inserted.  
committed.

An error message follows, indicating a unique constraint violation:

Error starting at line 206 in command:  
insert into trvl\_adv\_exp\_Details\_pbt (exp\_id,event\_id,proj\_id,emp\_id,exp\_type,exp\_cost,exp\_Desc)  
values  
(15, 'E110',1007,7003,'Lunch',150,'Lunch Details')  
Error report:  
SQL Error: ORA-00001: unique constraint (ORA\_PTAMHANKAR.SYS\_C0065259) violated  
00001. 00000 - "unique constraint (%s.%s) violated"  
\*Cause: An UPDATE or INSERT statement attempted to insert a duplicate key.  
For Trusted Oracle configured in DBMS MAC mode, you may see  
this message if a duplicate entry exists at a different level.  
\*Action: Either remove the unique restriction or do not insert the key.

A yellow callout box on the right side of the interface displays the user's information:

Name: Pratik Tamhankar  
Student Id: A20384093

## 4.3 Foreign Key check by inserting invalid EVENT ID

```

insert into trvl_adv_exp_Details_pbt (exp_id,event_id,proj_id,emp_id,exp_type,exp_cost,exp_Desc)
values
(16, 'E111',1099,7003,'Lunch',150,'Lunch Details');
commit;

```

Query Result x | Query Result 1 x | Script Output x | Query Result 3 x | Query Result 4 x | Query Result 5 x

Task completed in 0.347 seconds

Error starting at line 206 in command:

```

insert into trvl_adv_exp_Details_pbt (exp_id,event_id,proj_id,emp_id,exp_type,exp_cost,exp_Desc)
values
(16, 'E111',1099,7003,'Lunch',150,'Lunch Details')

```

Error report:

SQL Error: ORA-02291: integrity constraint (ORA\_PTAMHANKAR.FK\_PROJ\_ID1) violated - parent key not found  
02291. 00000 - "integrity constraint (%s.%s) violated - parent key not found"  
\*Cause: A foreign key value has no matching primary key value.  
\*Action: Delete the foreign key or add a matching primary key.

+ ... Name: Pratik Tamhankar  
Student Id: A20384093

#### UPDATE Anomalies:

Consider the below data in EMPLOYEE table, which has two employees with same name, RIA

	EMP_ID	EMP_NAME	EMP_DEPT_ID	EMP_DESIGNATION	EMP_PHONE	EMP_HIRE_DATE	EMP_MGR_ID
1	9999	TEST	5	TEST	1234567890	08-JAN-17	(null)
2	3001	ARJUN	2	PROJECT MANAGER HEAD	3121111813	01-JUN-15	3001
3	3002	YASH	2	PROJECT MANAGER	3121111823	01-JUN-16	3001
4	3003	ZEUS	2	PROJECT MANAGER	3121111834	01-JUL-16	3001
5	9001	LILLY	3	ACCOUNTING HEAD	7127111889	13-AUG-16	9001
6	9002	RIA	3	ACCOUNTANT	3127879099	12-MAY-16	9001
7	9003	RIA	3	ACCOUNTANT	3127879022	11-MAR-16	9001
8	7001	STATHAM	1	SALES HEAD	5123568867	08-JUL-15	7001
9	7002	JASON	1	SALES REPRESENTATIVE	5123568876	18-SEP-16	7001
10	7003	JESSICA	1	SALES REPRESENTATIVE	5123568855	11-AUG-16	7001
11	7004	JOHN	1	SALES REPRESENTATIVE	5123568850	19-SEP-16	7001

+ ... Name: Pratik Tamhankar  
Student Id: A20384093

Now, consider that the phone number of RIA with hire date 12-MAY-16 needs to be updated, and if one writes below update query,

**Query:**

```
update trvl_adv_employees_pbt emp set emp_phone=11111111110 where emp_name='RIA';
```

**Output:**

The screenshot shows the Oracle SQL Developer interface with several tabs at the top: Query Result 1, Script Output, Query Result 3, Query Result 4, and Query Result 5. Below the tabs, there are icons for Refresh, Undo, Redo, and SQL, followed by the message "All Rows Fetched: 11 in 0.084 seconds". The main area displays a table with 11 rows of employee data. A tooltip box is overlaid on the row for employee ID 9002, name RIA, who is a 3 ACCOUNTANT. The tooltip contains the text: "Name: Pratik Tamhankar" and "Student Id: A20384093".

	EMP_ID	EMP_NAME	EMP_DEPT_ID	EMP_DESIGNATION	EMP_PHONE	EMP_HIRE_DATE	EMP_MGR_ID
1	9999	TEST	5	TEST	1234567890	08-JAN-17	(null)
2	3001	ARJUN	2	PROJECT MANAGER HEAD	3121111813	01-JUN-15	3001
3	3002	YASH	2	PROJECT MANAGER	3121111823	01-JUN-16	3001
4	3003	ZEUS	2	PROJECT MANAGER	3121111834	01-JUL-16	3001
5	9001	LILLY	3	ACCOUNTING HEAD	7127111889	13-AUG-16	9001
6	9002	RIA	3	ACCOUNTANT	11111111110	12-MAY-16	9001
7	9003	RIA	3	ACCOUNTANT	11111111110	11-MAR-16	9001
8	7001	STATHAM	1	SALES HEAD	5123568867	08-JUL-15	7001
9	7002	JASON	1	SALES REPRESENTATIVE	5123568876	18-SEP-16	7001
10	7003	JESSICA	1	SALES REPRESENTATIVE	5123568855	11-AUG-16	7001
11	7004	JOHN	1	SALES REPRESENTATIVE	5123568850	19-SEP-16	7001

From the above Output its visible that all the EMPLOYEE records with the Name as “RIA” are updated due to the above query:

Hence, instead of the above query, one should always use the primary key field as a predicate condition for writing an update on such dirty data:

**Changed Query:**

```
update trvl_adv_employees_pbt emp
set emp_phone=11111111110 where emp_name='RIA' and emp_id=9002;
```

**Output:**

```
select * from trvl_adv_employees_pbt;
```

Query Result 1 | Script Output | Query Result 3 | Query Result 4 | Query Result 5

SQL | All Rows Fetched: 11 in 0.084 seconds

	EMP_ID	EMP_NAME	EMP_DEPT_ID	EMP_DESIGNATION	EMP_PHONE	EMP_HIRE_DATE	EMP_MGR_ID
1	9999	TEST	5	TEST	1234567890	08-JAN-17	(null)
2	3001	ARJUN	2	PROJECT MANAGER HEAD	3121111813	01-JUN-15	3001
3	3002	YASH	2	PROJECT MANAGER	3121111823	01-JUN-16	3001
4	3003	ZEUS	2	PROJECT MANAGER	3121111834	01-JUL-16	3001
5	9001	LILLY	3	ACCOUNTING HEAD	7127111889	13-AUG-16	9001
6	9002	RIA	3	ACCOUNTANT	11111111110	12-MAY-16	9001
7	9003	RIA	3	ACCOUNTANT	3127879022	11-MAR-16	9001
8	7001	STATHAM	1	SALES HEAD	5123568867	08-JUL-15	7001
9	7002	JASON	1	SALES REPRESENTATIVE	5123568876	18-SEP-16	7001
10	7003	JESSICA	1	SALES REPRESENTATIVE	5123568855	11-AUG-16	7001
11	7004	JOHN	1	SALES REPRESENTATIVE	5123568850	19-SEP-16	7001

+ ...

Name: Pratik Tamhankar  
Student Id: A20384093

Hence, by using appropriate keys in the Updates performed, Update Anomalies can be avoided

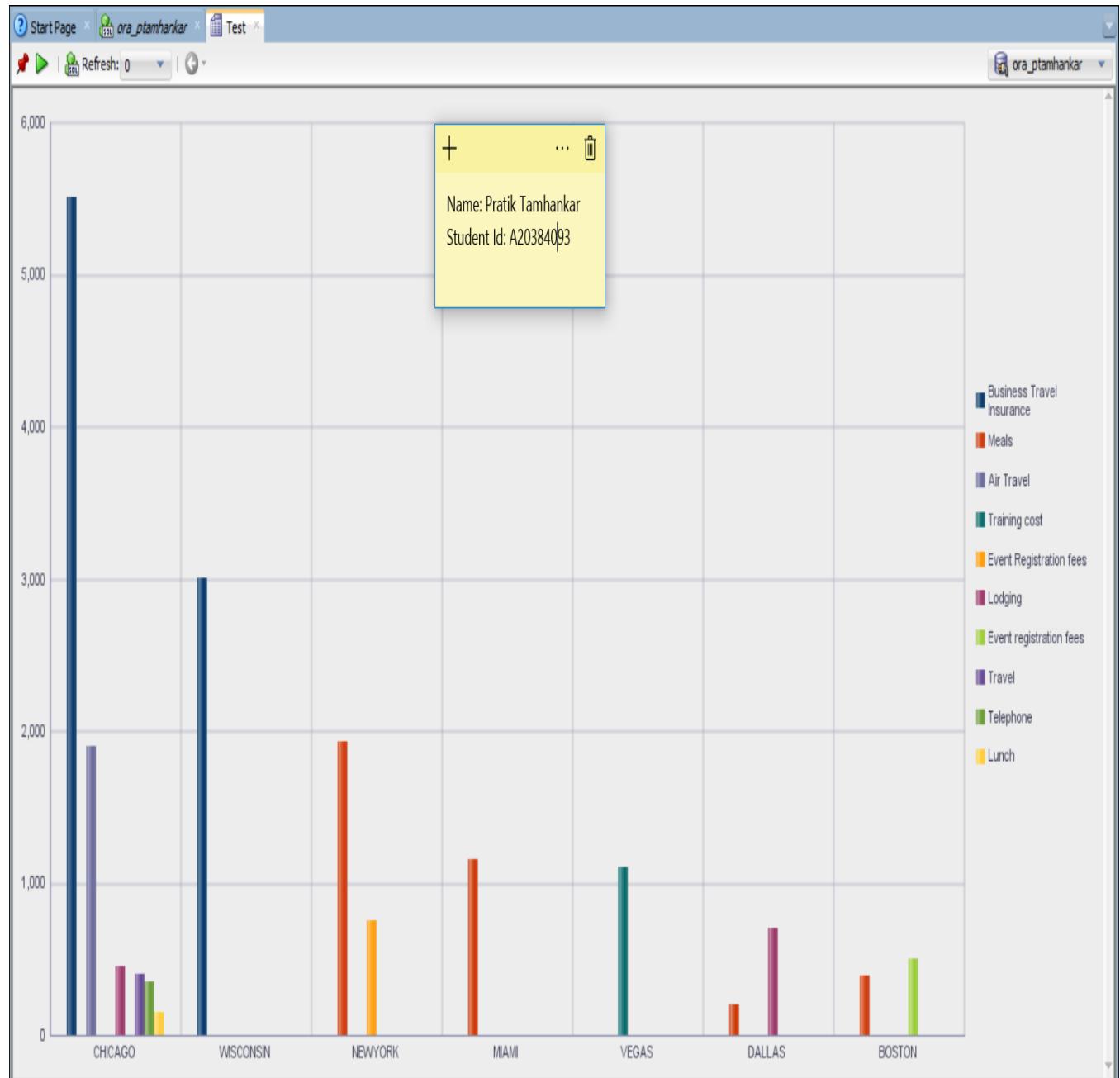
## PHASE IX

### Data Analytics Performed

1: Location wise expense areas

#### Query:

```
select EVENT_LOC,EXP_TYPE,expenses from
(
SELECT I.EVENT_STATUS,
I.EVENT_DESC,
I.EVENT_LOC,
J.EXP_TYPE,
sum(J.EXP_COST) expenses
FROM TRVL_ADV_EXP_DETAILS_PBT J,
TRVL_ADV_EVENTS_PBT I
WHERE I.EVENT_ID = J.EVENT_ID
group by
I.EVENT_STATUS,
I.EVENT_DESC,
I.EVENT_LOC,
J.EXP_TYPE
) order by expenses desc
```

**Output:**

2: Project Wise Number of requests made, current project cost and current request amounts

**Query:**

```
select proj.PROJ_NAME,proj.proj_cost,count(req.TA_REQUEST_ID)
requests_made ,sum(req.requested_amount) total_request_amount
from trvl_adv_request_pbt req,trvl_adv_proj_pbt proj
where req.proj_id=proj.PROJ_ID
group by proj.PROJ_NAME,proj.proj_cost;
```

**Output:**

The screenshot shows a SQL query interface with the following details:

Query:

```
select proj.PROJ_NAME,proj.proj_cost,count(req.TA_REQUEST_ID) requests_made ,sum(req.requested_amount) total_request_amount
from trvl_adv_request_pbt req,trvl_adv_proj_pbt proj
where req.proj_id=proj.PROJ_ID
group by proj.PROJ_NAME,proj.proj_cost;
```

Result:

Query Result | All Rows Fetched: 6 in 0.047 seconds

PROJ_NAME	PROJ_COST	REQUESTS_MADE	TOTAL_REQUEST_AMOUNT
1 Project Loon	28000	2	890
2 Project Tango	75000	1	5500
3 Project Ara	75000	2	2647
4 Waymo	75000	1	900
5 Accounting reconciliation framework	75000	4	5350
6 Robo Army	43000	2	3025

Details:

Name: Pratik Tamhankar  
Student Id: A20384093

3:

**Query:**

```
select event.event_Desc,sum(settle.set_total_Expenses) "Expenses",sum(settle.set_total_Advances) as "Advances" from  
trvl_adv_settle_details_pbt settle,trvl_adv_events_pbt event  
where settle.set_Event_id=event.event_id  
group by event.event_Desc;
```

**Output:**

The screenshot shows the Oracle SQL Developer interface. At the top, there is a code editor window containing the SQL query. Below it is a toolbar with icons for Query Result, Refresh, Stop, and SQL. The message 'All Rows Fetched: 7 in 0.051 seconds' is displayed. To the right of the toolbar is a yellow callout box containing the student's name and ID. The main area shows a table with columns EVENT\_DESC, Expenses, and Advances, listing seven rows of data.

EVENT_DESC	Expenses	Advances
1 Pre-Sales Meet	5197	3997
2 Customer Meet	3000	3000
3 Sales meetup	5500	5500
4 Contract bid meet	1780	890
5 Salers meet	3025	3025
6 Sales Bid Meet	750	750
7 Infotech product line launch	1150	1150

4:

**Query:**

```
SELECT
A.EMP_ID,
A.EMP_NAME,
C.APPROVAL_STATUS,
D.TA_STATUS,
D.TA_REMITTANCE_TYPE,
count(D.TA_ISSUANCE_ID) issued_count,
count(B.TA_REQUEST_ID) request_Count,
sum(B.REQUESTED_AMOUNT) total_request_Amount
FROM TRVL_ADV_REQUEST_PBT B,
TRVL_ADV_EMPLOYEES_PBT A,
TRVL_ADV_RESPONSE_PBT C,
TRVL_ADV_TRAVEL_ADVANCES_PBT D
WHERE A.EMP_ID = B.REQUESTER_ID
AND B.TA_REQUEST_ID = C.TA_REQ_ID
AND C.TA_RESP_ID = D.TA_RESPONSE_ID
group by
A.EMP_ID,
A.EMP_NAME,
B.TA_APPORVAL_EMP_ID,
C.APPROVAL_STATUS,
D.TA_STATUS,
D.TA_REMITTANCE_TYPE
order by sum(B.REQUESTED_AMOUNT) desc
```

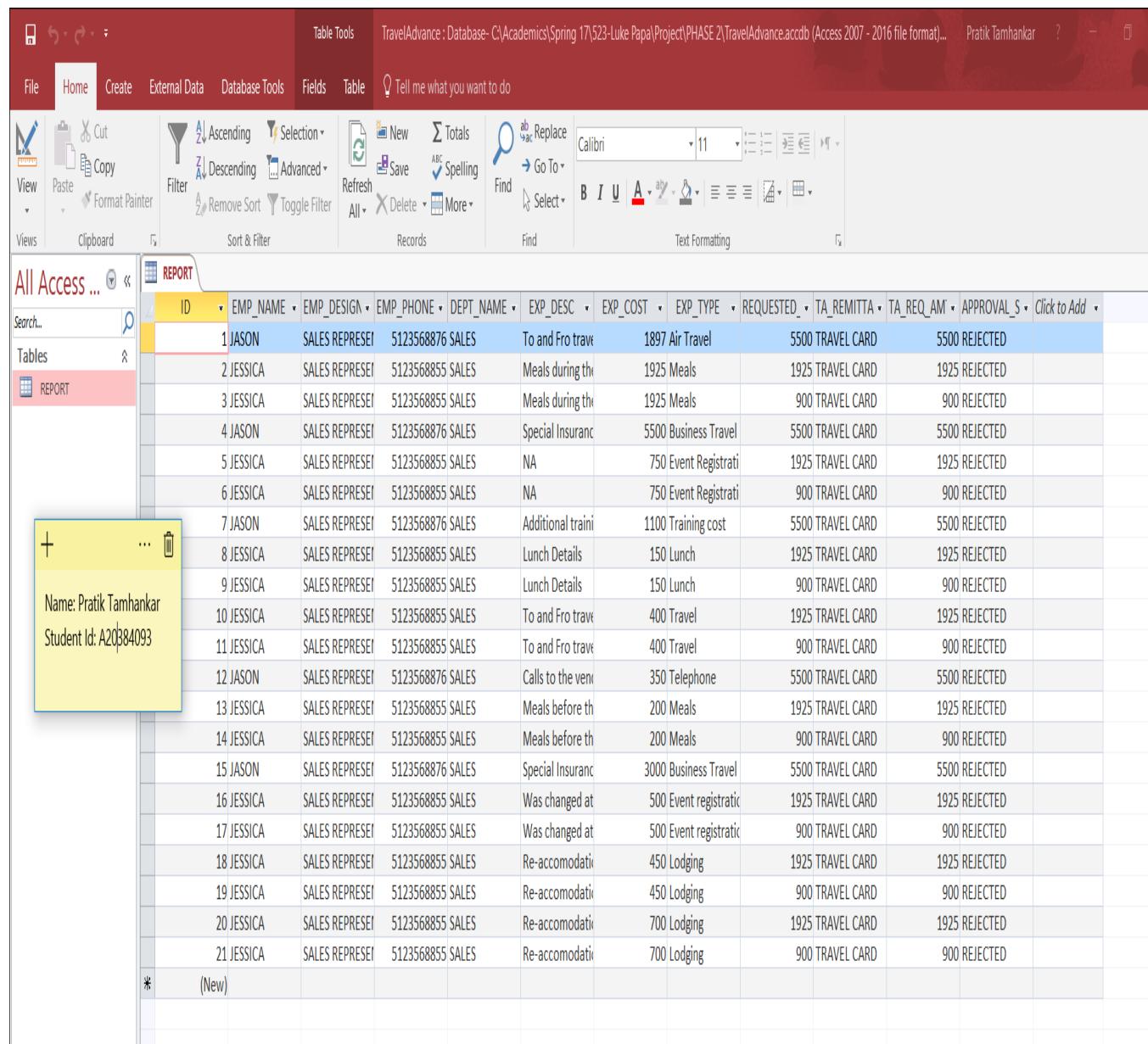
**Output:**

The screenshot shows the Oracle SQL Developer interface. At the top, there are three tabs: 'Query Result' (highlighted), 'Query Result 1', and 'Query Result 2'. Below the tabs, there are icons for refresh, undo, redo, and a red error icon, followed by the text 'SQL | All Rows Fetched: 5 in 0.031 seconds'. The main area displays a table with the following data:

E...	Y...	EMP_NAME	APPROVAL_STATUS	TA_STATUS	TA_REMITTANCE_TYPE	ISSUED_COUNT	REQUEST_COUNT	TOTAL_REQUEST_AMOUNT
1	7002	JASON	APPROVED	OPEN	CASH	3	3	5997
2	7004	JOHN	APPROVED	OPEN	TRAVEL CARD	2	2	1540
3	7003	JESSICA	APPROVED	OPEN	CASH	2	2	1350
4	7003	JESSICA	APPROVED	OPEN	TRAVEL CARD	1	1	750
5	7002	JASON	APPROVED	OPEN	TRAVEL CARD	1	1	350

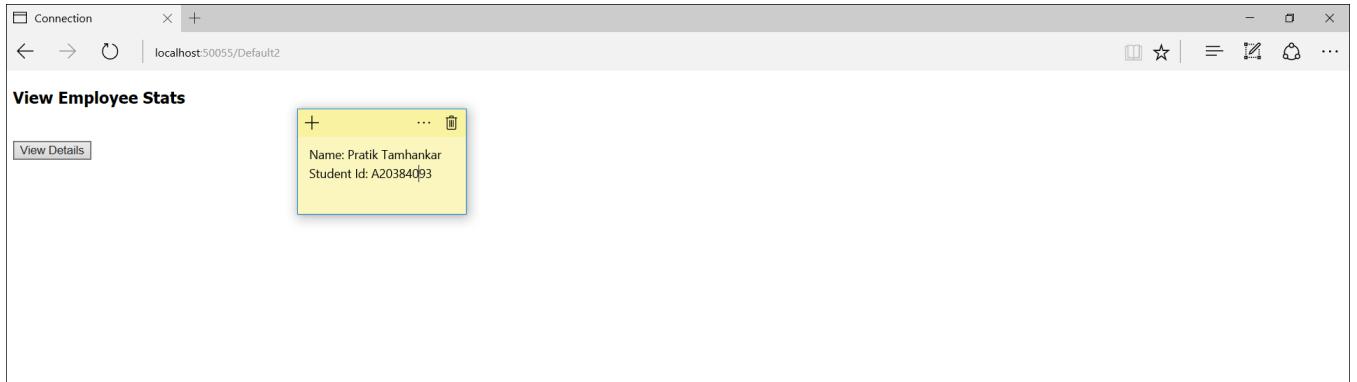
A yellow callout box is positioned over the last row of the table, containing the text: 'Name: Pratik Tamhankar' and 'Student Id: A20384093'.

## WEB Presence

**1: Access Database Table**


The screenshot shows the Microsoft Access application interface. The ribbon is visible at the top with tabs for File, Home, Create, External Data, Database Tools, Fields, Table, and Table Tools. The Table Tools tab is selected. The main area displays a report view of a table named "TravelAdvance". The table has columns: ID, EMP\_NAME, EMP\_DESIGN, EMP\_PHONE, DEPT\_NAME, EXP\_DESC, EXP\_COST, EXP\_TYPE, REQUESTED\_DATE, TA\_REMITTANCE, TA\_REQ\_AMOUNT, APPROVAL\_STATUS, and Click to Add. The data shows various travel entries for employees like Jason and Jessica, including details like meal costs, lodging, and travel type. A yellow callout box highlights the first row for Jason.

ID	EMP_NAME	EMP_DESIGN	EMP_PHONE	DEPT_NAME	EXP_DESC	EXP_COST	EXP_TYPE	REQUESTED_DATE	TA_REMITTANCE	TA_REQ_AMOUNT	APPROVAL_STATUS	Click to Add
1 JASON	SALES REPRESE	15123568876	SALES	To and Fro trav	1897 Air Travel	5500	TRAVEL CARD	5500	REJECTED			
2 JESSICA	SALES REPRESE	15123568855	SALES	Meals during th	1925 Meals	1925	TRAVEL CARD	1925	REJECTED			
3 JESSICA	SALES REPRESE	15123568855	SALES	Meals during th	1925 Meals	900	TRAVEL CARD	900	REJECTED			
4 JASON	SALES REPRESE	15123568876	SALES	Special Insuranc	5500 Business Travel	5500	TRAVEL CARD	5500	REJECTED			
5 JESSICA	SALES REPRESE	15123568855	SALES	NA	750 Event Registrati	1925	TRAVEL CARD	1925	REJECTED			
6 JESSICA	SALES REPRESE	15123568855	SALES	NA	750 Event Registrati	900	TRAVEL CARD	900	REJECTED			
7 JASON	SALES REPRESE	15123568876	SALES	Additional traini	1100 Training cost	5500	TRAVEL CARD	5500	REJECTED			
8 JESSICA	SALES REPRESE	15123568855	SALES	Lunch Details	150 Lunch	1925	TRAVEL CARD	1925	REJECTED			
9 JESSICA	SALES REPRESE	15123568855	SALES	Lunch Details	150 Lunch	900	TRAVEL CARD	900	REJECTED			
10 JESSICA	SALES REPRESE	15123568855	SALES	To and Fro trav	400 Travel	1925	TRAVEL CARD	1925	REJECTED			
11 JESSICA	SALES REPRESE	15123568855	SALES	To and Fro trav	400 Travel	900	TRAVEL CARD	900	REJECTED			
12 JASON	SALES REPRESE	15123568876	SALES	Calls to the ven	350 Telephone	5500	TRAVEL CARD	5500	REJECTED			
13 JESSICA	SALES REPRESE	15123568855	SALES	Meals before th	200 Meals	1925	TRAVEL CARD	1925	REJECTED			
14 JESSICA	SALES REPRESE	15123568855	SALES	Meals before th	200 Meals	900	TRAVEL CARD	900	REJECTED			
15 JASON	SALES REPRESE	15123568876	SALES	Special Insuranc	3000 Business Travel	5500	TRAVEL CARD	5500	REJECTED			
16 JESSICA	SALES REPRESE	15123568855	SALES	Was changed at	500 Event registratio	1925	TRAVEL CARD	1925	REJECTED			
17 JESSICA	SALES REPRESE	15123568855	SALES	Was changed at	500 Event registratio	900	TRAVEL CARD	900	REJECTED			
18 JESSICA	SALES REPRESE	15123568855	SALES	Re-accomodati	450 Lodging	1925	TRAVEL CARD	1925	REJECTED			
19 JESSICA	SALES REPRESE	15123568855	SALES	Re-accomodati	450 Lodging	900	TRAVEL CARD	900	REJECTED			
20 JESSICA	SALES REPRESE	15123568855	SALES	Re-accomodati	700 Lodging	1925	TRAVEL CARD	1925	REJECTED			
21 JESSICA	SALES REPRESE	15123568855	SALES	Re-accomodati	700 Lodging	900	TRAVEL CARD	900	REJECTED			
*	(New)											

**2: Web Output:**

A screenshot of a web browser window titled "View Employee Stats". A modal dialog box is displayed, containing the text: "Name: Pratik Tamhankar" and "Student Id: A20384093". Below the dialog box is a table with 15 rows of data. The table has a yellow header row and a blue border.

EMP_NAME	EMP_DESIGNATION	EMP_PHONE	DEPT_NAME	EXP_DESC	EXP_COST	EXP_TYPE	REQUESTED_AMOUNT	TA_REMITTANCE_TYPE	TA_REQ_AMT
JASON	SALES REPRESENTATIVE	5123568876	SALES	To and Fro travel till event location and back	1897	Air Travel	5500	TRAVEL CARD	5500
JESSICA	SALES REPRESENTATIVE	5123568855	SALES	Meals during the event, before the event and after the event	1925	Meals	1925	TRAVEL CARD	1925
JESSICA	SALES REPRESENTATIVE	5123568855	SALES	Meals during the event, before the event and after the event	1925	Meals	900	TRAVEL CARD	900
JASON	SALES REPRESENTATIVE	5123568876	SALES	Special Insurance as per the location was required	5500	Business Travel Insurance	5500	TRAVEL CARD	5500
JESSICA	SALES REPRESENTATIVE	5123568855	SALES	NA	750	Event Registration fees	1925	TRAVEL CARD	1925
JESSICA	SALES REPRESENTATIVE	5123568855	SALES	NA	750	Event Registration fees	900	TRAVEL CARD	900
JASON	SALES REPRESENTATIVE	5123568876	SALES	Additional training cost wasn't included in Original Event Itinerary	1100	Training cost	5500	TRAVEL CARD	5500
JESSICA	SALES REPRESENTATIVE	5123568855	SALES	Lunch Details	150	Lunch	1925	TRAVEL CARD	1925
JESSICA	SALES REPRESENTATIVE	5123568855	SALES	Lunch Details	150	Lunch	900	TRAVEL CARD	900
JESSICA	SALES REPRESENTATIVE	5123568855	SALES	To and Fro travel till airport, event location and back	400	Travel	1925	TRAVEL CARD	1925
JESSICA	SALES REPRESENTATIVE	5123568855	SALES	To and Fro travel till airport, event location and back	400	Travel	900	TRAVEL CARD	900
JASON	SALES REPRESENTATIVE	5123568876	SALES	Calls to the vendors overseas	350	Telephone	5500	TRAVEL CARD	5500
JESSICA	SALES REPRESENTATIVE	5123568855	SALES	Meals before the event date weren't included	200	Meals	1925	TRAVEL CARD	1925

**3: ASP.NET Code:**

```
<%@ Page Language = "VB" %>
<%@ Import Namespace = "System.Data.OleDb" %>
<!DOCTYPE html>
<html xmlns = "http://www.w3.org/1999/xhtml">
<head id="Head1" runat = "server">
<title>Connection</title>
<script runat = "server">

Sub Search_Click(Src As Object, E As EventArgs)
    Try
        'Connect to the Database
        Dim cnAccess As New OleDbConnection(
            "Provider = Microsoft.Jet.OLEDB.4.0;" &
            "Data Source = C:\Academics\Spring 17\523-Luke Papa\Project\Test\TA.mdb")

        cnAccess.Open()

        'Construct the SELECT statement

        Dim sSelectSQL As String
        'Create the SQL Select Statement

        sSelectSQL = "SELECT * FROM REPORT"

        'Create the OleDbCommand object
        Dim cmdSelect As New OleDbCommand(sSelectSQL, cnAccess)
        Dim drEmp As OleDbDataReader, sbResults As New StringBuilder()

        drEmp = cmdSelect.ExecuteReader()
        sbResults.Append("<table border=1>")
        sbResults.Append("</td><td>")
        sbResults.Append(drEmp.GetName(1))
        sbResults.Append("</td><td>")
        sbResults.Append(drEmp.GetName(2))
        sbResults.Append("</td><td>")
        sbResults.Append(drEmp.GetName(3))
        sbResults.Append("</td><td>")
        sbResults.Append(drEmp.GetName(4))
        sbResults.Append("</td><td>")
        sbResults.Append(drEmp.GetName(5))
        sbResults.Append("</td><td>")
        sbResults.Append(drEmp.GetName(6))
        sbResults.Append("</td><td>")
        sbResults.Append(drEmp.GetName(7))
        sbResults.Append("</td><td>")
        sbResults.Append(drEmp.GetName(8))
        sbResults.Append("</td><td>")
        sbResults.Append(drEmp.GetName(9))
        sbResults.Append("</td><td>")
```

```
sbResults.Append(drEmp.GetName(10))
sbResults.Append("</td>")
sbResults.Append("</tr>")

Do While drEmp.Read()

    sbResults.Append("<tr><td>")
    sbResults.Append(drEmp.GetString(1))
    sbResults.Append("</td><td>")
    sbResults.Append(drEmp.GetString(2))
    sbResults.Append("</td><td>")
    sbResults.Append(drEmp.GetDouble(3).ToString)
    sbResults.Append("</td><td>")
    sbResults.Append(drEmp.GetString(4))
    sbResults.Append("</td><td>")
    sbResults.Append(drEmp.GetString(5))
    sbResults.Append("</td><td>")
    sbResults.Append(drEmp.GetDouble(6).ToString)
    sbResults.Append("</td><td>")
    sbResults.Append(drEmp.GetString(7))
    sbResults.Append("</td><td>")
    sbResults.Append(drEmp.GetDouble(8).ToString)
    sbResults.Append("</td><td>")
    sbResults.Append(drEmp.GetString(9))
    sbResults.Append("</td><td>")
    sbResults.Append(drEmp.GetDouble(10).ToString)
    sbResults.Append("</td></tr>")

Loop
sbResults.Append("</table>")
msg.Text = sbResults.ToString()

cnAccess.Close()
cnAccess = Nothing
Catch ex As Exception
    Response.Write(ex.Message)
    Response.Write("Connection Failed")
End Try
End Sub

</script>
</head>
<body style = "font-family:Tahoma;">
<h3>View Employee Stats</h3>
<form runat = "server" id = "form1">
<br />
<asp:Button Text = "View Details" OnClick = "Search_Click"
runat = "server" ID = "Button1" />
<p>
<asp:Label id = "msg" runat = "server" />
</p>
</form>
<div></div>
</body>
</html>
```

## PHASE X

### Systems Analysis and Viewpoints

#### 1: Business Intelligence:

BI refers to analyzing data and presenting actionable information to help corporate executives, business managers and other end users make more informed business decisions. BI encompasses a wide variety of tools, applications and methodologies that enable organizations to collect data from internal systems and external sources, prepare it for analysis, develop and run queries against the data, and create reports, dashboards and data visualizations to make the analytical results available to corporate decision makers as well as operational workers.

This project provides physical models in a way that flexible BI reports and intelligent charts can be viewed and created dynamically

#### 2: Data Management:

Objectives: Reduced Normalization and Anomalies

The physical model provided in this project has minimal redundancies and minimum update, insert and delete anomalies

#### 3: Predictive Analytics:

This refers to predicting of future data based on the current data.

This project provides the set of data structures to perform such analysis and ensures that system scalability is achievable

#### 4: Data Gravity

Data gravity is an analogy of the nature of data and its ability to attract additional applications and services. The Law of Gravity states that the attraction between objects is directly proportional to their weight (or mass).