Shah, Sunny
Dixit, Rushikesh

09/29/16

CSE 5330 – Database Systems (Fall 2016)

Foreign Academic & Consulting Service

Query 1:

--system determines which month has maximum
--number of enquiries and enrollment with the firm.

```
select month_name,
nvl(temp_total_enquiries,0) as total_enquiries,
total_enrolls
from (select monthname,
count(enquiries) as temp_total_enquiries
from (select to_char(dateofEnquiry, 'Month') as monthname,
dateofEnquiry as enquiries
from F16_12_enquiresAt)
group by monthname),
(select month_name as month_name,
count(appl_id) as total_enrolls
from (select enroll.applicantId as appl_id,
to_char(enroll.applicantdoj , 'Month') as month_name
from f16_12_enrollsIn enroll)
group by month_name)
where monthname(+) = month_name;
```

MONTH_NAME	TOTAL_ENQUIRIES	TOTAL_ENROLLS
September	10	5
July	1	5
June	1	6
April	2	6
November	5	5
May	1	6
October	10	5
August	2	5
March	3	7
January	0	8
February	0	7
December	0	3

12 rows selected.

Query 2:

--system compares which locations or areas the
--enquiries and enrollment received is maximum

```
select nvl(enquirer_state,applicant_state) as cust_state,
enquirer_count,
applicant_count
from (select cust.state as enquirer_state,
count(enq.enquirerID) as enquirer_count
from F16_12_customer cust,
F16_12_enquirer enq
where cust.customerID = enq.enquirerID
group by state) full outer join
(select cust.state as applicant_state,
count(app.applicantID) as applicant_count
from F16_12_customer cust,
F16_12_applicant app
where cust.customerID = app.applicantID
group by state)
on enquirer_state = applicant_state;
```

CUST_STATE	ENQUIRER_COUNT	APPLICANT_COUNT
WV	1	1
AK	2	2
MN	1	–
CA	2	3
AL	2	1
ND	1	–
MA	1	–
VA	1	1
NE	2	–
NH	1	1
MI	1	–
GA	1	1
SC	2	1
RI	2	–
DC	2	–
MT	1	1

MS	1	–
LA	1	1
IN	2	–
HI	1	1
NV	1	1
TX	2	–
OR	2	–
PA	1	1
AZ	1	–
WA	–	1
KS	–	1
MD	–	1
ID	–	1
OH	–	2
VT	–	1

31 rows selected.

Query 3:

--Based on customer enquiries and applicants system determines
--which country is more popular or has a high preference for further details

```
select enq_country,
total_enquirer,
nvl(total_app,0) as total_applicant
from (select enq.countryPref as enq_country,
count(enq.enquirerID) as total_enquirer
from F16_12_enquirer enq
group by enq.countryPref
order by total_enquirer) full outer join
(select app.countryPref as app_country,
count(app.applicantID) as total_app
from F16_12_applicant app
group by app.countryPref
order by total_app)
on enq_country = app_country
order by total_applicant desc;
```

ENQ_COUNTRY	TOTAL_ENQUIRER	TOTAL_APPLICANT
Germany	4	7
Canada	4	4
Singapore	3	4
USA	5	3
United Kingdom	4	2
Italy	4	2
Bangladesh	1	1
Lesotho	1	0
Oman	1	0
Burundi	1	0
Ireland	1	0
Tokelau	1	0
Argentina	1	0
Sao Tome and Principe	1	0
Tunisia	1	0
Serbia and Montenegro	1	0
Czech Republic	1	0

17 rows selected.

Query 4:

--Generating income reports of assistance services as to determine
--which services are generating more revenue

```
select * from (select *
from (select service.serviceType,
sum(service.serviceFee) as total_fee
from f16_12_service service
group by service.serviceType)
order by total_fee desc);
```

SERVICETYPE	TOTAL_FEE
Tutoring	360000
Visa Assist	260000
University Applications	126000

3 rows selected.

Query 5:

--Displaying profit of the firm with respect to branches

```
select branch_id, income - expense as profit
from (select branch_id,
      sum(service_fee) as income
        from (select br.branchID as branch_id,
                br.serviceID branch_service_id
                from F16_12_enrollsIn br
                order by serviceID
             ), (select serv.serviceID as service_id,
            serv.serviceFee as service_fee
            from F16_12_service serv
        )
        where branch_id = service_id
        group by branch_id),
        (select sum(monthly_salary) as expense
        from F16_12_employee
        group by branch)
group by branch_id;
```

BRANCH_ID	PROFIT
1	176500
2	176500

2 rows selected.

Query 6:

--System generates performance outline of tutors by applicant rating

```
select DISTINCT emp_no, emp_name, rating
from (select emp_no, emp_name, rating
        from (select em.employeeid as emp_no,em.firstname as emp_name
        from F16_12_employee em),
            (select r.employeeid emp_no_rating, avg(r.rating) as rating
            from F16_12_rates r
            group by r.employeeid)
       where emp_no = emp_no_rating),
       (select enroll.employeeID emp, serv.servicetype
         from F16_12_enrollsin enroll, F16_12_service serv
       where serv.servicetype='Tutoring' and serv.serviceID = enroll.serviceID)
where emp_no = emp;
```

EMP_NO	EMP_NAME	RATING
20150019	Bree	4
20150016	Howard	4
20150018	Danielle	5
20150017	Iris	4.5
20150022	Chaney	4
20150021	Kato	3
20150023	Roary	4
20150020	Petra	5
20150024	Malachi	5

9 rows selected.

Query 7:

--Performance report of counselors based on ratings of the applicant

```
select  emp_no, emp_name, rating
from (select emp_no, emp_name, rating
       from (select em.employeeid as emp_no,em.firstname as emp_name
       from F16_12_employee em),
          (select r.employeeid emp_no_rating, avg(r.rating) as rating
          from F16_12_rates r
          group by r.employeeid)
      where emp_no = emp_no_rating),
      (select emp.employeeID empl, emp.designation
        from F16_12_employee emp
      where emp.designation='Counselor')
where emp_no = empl;
```

EMP_NO	EMP_NAME	RATING
20150012	Lysandra	4
20150011	Yuli	4
20150013	Lacy	4
20150008	Conan	4.333333333333333333333333333333333333
20150009	Hanna	3.333333333333333333333333333333333333
20150010	Candice	4

6 rows selected.

Query 8:

--Generating reports of tutor class popularity that is which course class has more popularity

```
select em.firstname || ' ' || em.lastname as Tutor_Name, tu.classtype, count(en.applicantid) as
Total
from f16_12_enrollsin en, f16_12_tutorclass tu, f16_12_employee em
where en.serviceid=tu.serviceid
group by em.firstname,em.lastname, tu.classtype;
```

TUTOR_NAME	CLASSTYPE	TOTAL
Dean Donovan	GRE	10
Ahmed Livingston	IELTS	8
Hanna Hyde	GRE	10

Ginger Williamson	IELTS	8
Kelly Sampson	TOEFL	6
Howard Vaughan	TOEFL	6
Bree Trujillo	GRE	10
Kato Browning	GRE	10
Roary Lowe	TOEFL	6
Roary Lowe	GRE	10
Dean Donovan	TOEFL	6
Geraldine Travis	IELTS	8
Ahmed Livingston	GRE	10
Martina Jones	IELTS	8
Lysandra Scott	GRE	10
Chaney Cleveland	GRE	10
Geraldine Travis	GRE	10
Brent Cummings	TOEFL	6
Martina Jones	TOEFL	6
Felix Newman	GRE	10
Conan Gay	IELTS	8
Hanna Hyde	IELTS	8
Yuli Hartman	TOEFL	6
Ginger Williamson	GRE	10
Howard Vaughan	IELTS	8
Petra Mack	GRE	10
Chaney Cleveland	IELTS	8
Ahmed Livingston	TOEFL	6
Hanna Hyde	TOEFL	6
Yuli Hartman	IELTS	8
Lacy Holloway	GRE	10
Lacy Holloway	IELTS	8
Kelly Sampson	IELTS	8

Iris Saunders	TOEFL	6
Danielle Welch	TOEFL	6
Danielle Welch	IELTS	8
Kato Browning	TOEFL	6
Chaney Cleveland	TOEFL	6
Felix Newman	TOEFL	6
Bree Trujillo	TOEFL	6
Malachi Parker	TOEFL	6
Fredericka Buchanan	TOEFL	6
Brent Cummings	GRE	10
Conan Gay	TOEFL	6
Yuli Hartman	GRE	10
Lysandra Scott	TOEFL	6
Kelly Sampson	GRE	10
Iris Saunders	GRE	10
Bree Trujillo	IELTS	8
Roary Lowe	IELTS	8
Malachi Parker	GRE	10
Dean Donovan	IELTS	8
Brent Cummings	IELTS	8
Conan Gay	GRE	10
Candice Mcclure	TOEFL	6
Lacy Holloway	TOEFL	6
Ginger Williamson	TOEFL	6
Howard Vaughan	GRE	10
Kato Browning	IELTS	8
Fredericka Buchanan	GRE	10
Fredericka Buchanan	IELTS	8
Geraldine Travis	TOEFL	6
Martina Jones	GRE	10

Felix Newman	IELTS	8
Candice Mcclure	GRE	10
Candice Mcclure	IELTS	8
Lysandra Scott	IELTS	8
Iris Saunders	IELTS	8
Danielle Welch	GRE	10
Petra Mack	TOEFL	6
Petra Mack	IELTS	8
Malachi Parker	IELTS	8

72 rows selected.

Query 9:

--Displaying the VISA success or acceptance ratio

```
select 'ACCEPT:TOTAL' as description,
success_count || ':' || total_record as acceptance_rate
from (select count(en.applicantid) as success_count
from f16_12_enrollsin en, f16_12_visarecord v
where en.serviceid=v.serviceid
and v.visastatus = 'accept'
group by v.visastatus),
(select count(*) as total_record
from f16_12_visarecord);
```

DESCRIPTION	ACCEPTANCE_RATE
ACCEPT:TOTAL	14:26