**B: SQL**

1) What does *UNION*? What is the difference between *UNION* and *UNION ALL*? What does *INTERSECT* ? What are the prerequisites to use these statements?

-Union is the operator of the union of two or more sets (samples), the result of which is unique rows in sorted format.

-The difference between Union and Union All operators is that Union All displays all the sample values as a result.

-Intersect is a set intersection operator, the result of which are unique sorted lines.

Example: Consider we have two mobile clients(Client\_1 and Client\_2) with some mobile packages and we would like to get different results such as: Similar packages, Different packages, Client\_1’s packages, Clients\_2’s packages.

2) Explain shortly the idea of transaction blocks.

- The set of changes on the Database (Update, Insert or Delete).

-The executing of the transaction are operators (Commit, Rollback)

3)What does the statement *CREATE TRIGGER* ? Explain shortly.

-Before or after handlers over «Insert», «Update» or «Delete» or over database system events, for instance, «Log on» or «Log off»

Example: What does who do in Database(you can monitor this actions)

4)Write a statement to extract number of full months (result shown as integer) between two timestamps: *date1* and *date2*.

«select trunc(months\_between(to\_timestamp(sysdate+70),(to\_timestamp(sysdate)))) from dual»

5) Explain the results from the query below. What is the name of the function in the last column? What is the difference between this function and a regular aggregate function. How else can we use this function? Give some examples.

—————————

SELECT depname, empno, salary, rank() OVER (PARTITION BY depname ORDER BY salary

DESC) FROM empsalary;

—————————

-It will displays a table with the columns depname, empno, salary, “rank”. The result of the analytical function will be the numbers in order, built by groups from the depname field and sorted by the salary field in descending order.

-RANK - it’s an analytic function.

-The main difference between these two functions is that aggregate functions are used to return a single result value based on a group of rows. When we use the analytic function, we must use “OVER” and we also need to use order by or partition by as analytic

-As an aggregate function, "RANK" calculates the rank of a hypothetical row identified by the arguments of the function with respect to a given sort specification. The arguments of the function must all evaluate to constant expressions within each aggregate group, because they identify a single row within each group

6) Write a statement to create a multicolumn index on columns: loan\_id, bank\_id in loan\_request table.

create table loan\_request (loan\_id number , bank\_id number);

create index i\_loan\_request\_comp on loan\_request(loan\_id, bank\_id);

7) Write a statement to find all records from table person for which person\_id is NOT unique.

select \*

from person p

where p.preson\_id in (select p.preson\_id

from person p

group by preson\_id

having count(preson\_id) > 1)

8) Can you write a simple SQL statement which selects the last\_name and application\_date of customers who have applied after January 12th 2012, whereas:

-Column last\_name is in table person

-Column application\_date is in table account

-The connection between both tables is via the column account\_id which exists in both tables

select p.last\_name, a.application\_date

from person p, account a

where p.account\_id = a.account\_id

and to\_date(to\_char(a.application\_date, 'mm.dd.yyyy'), 'mm.dd.yyyy') >

to\_date('01.12.2012', 'mm.dd.yyyy')

9) There are given two tables (details below). Write ONE query to show for each customer: the last successful loan\_request (loan\_id, creation\_date, loan\_amount), sum of loan\_amount of all failed loan\_requests, ratio of successful\_to\_failed applications. (nice to have)

---the last successful loan\_request (loan\_id, creation\_date, loan\_amount)

select ca.account\_id,

lr1.loan\_id,

ext1.creation\_date,

lr1.loan\_amount

from loan\_request lr1,

(select max(lr.creation\_date) creation\_date, lr.account\_id

from loan\_request lr

group by lr.account\_id

where lr.loan\_amount = 'SUCCESS') ext1,

customer\_account ca

where ext1.creation\_date = lr1.creation\_date

and ext.account\_id = lr1.account\_id

and ca.account\_id = lr1.account\_id

and ext.account\_id = ca.account\_id

**С:R**

1) Can you write a R statement which gives you the average, median and variance of the variable main\_income

-mean(main\_icome)

-median(main\_income)

-sd(main\_income)

-var(main\_income)

OR

-summary(main\_income)

2) Please interpret the following statements

a. ba\_level\_postbank <- ba\_level\_postbank[,!(names(ba\_level\_postbank) == "signup\_date")]

-Create ba\_level\_postbank variable which will consist of all the rows from names column with not «signup\_value» value.

b. ba\_level\_postbank$ec\_property\_owned\_square\_metres[is.na(ba\_level\_postbank$ec\_proper ty\_owned\_square\_metres)] <- 0

-From the dataset ba\_level\_postbank choose column ec\_property\_owned\_square\_metres. And in this column values = NaN change into zeroes (0).

c. ba\_level\_postbank <- subset(ba\_level\_postbank, received\_palimony >= 0)

-Create ba\_level\_postbank variable which will consist of subset from all the dataset where in the column received\_polimony values will be >= 0

d. ba\_level\_postbank$lead\_canal <- as.factor(ba\_level\_postbank$lead\_canal)

-Change the type of lead\_canal column from ba\_level\_postbank dataset into format column

e. ba\_level\_postbank$effective\_interest <- as.factor(ifelse(ba\_level\_postbank$ effective\_interest <= 6, 1, ifelse(ba\_level\_postbank$ effective\_interest <= 8, 2, 3)))

**-Shortly:**

«if» value in effective\_interest <= 6 then 1 «elif» value in effective\_interest <= 8 then 2 «else» 3

**-Full answer:**

Create a variable ba\_level\_postbank with such values as IF in the column effective\_interest value <=6 (True) then value ‘1’ ELSE IF in the column effective\_interest value <=8 (True) then value ‘2’ ELSE value ‘3’