

(d)

Ans to the ques no : 01(c)

Declare

total numbers;

Begin

Update EMP

Set Salaryy = Salary + 0.1 * Salary

Where Salary < (Select avg(Salary)
From EMP);

If sql%notfound then

dbms_output.put_line ("NO RECORDS ARE UPDATED");

elsif sql%found then

total := sql%rowcount;

dbms_output.put_line ("A TOTAL OF" || total || " ARE UPDATED");

endif;

End;

/

(2)

Ans to the ques no: 1(a)

Bitmap indexing are useful when there is query for selecting multiple keys.

But when there is need for updating, bitmap indices take more time to be updated.

So, when there is need of update, bitmap indexing degrades performance.

Ans to the ques no: 1(d)

Declare

amount number;

v_date date;

Begin

Select amountPaid, DatePaid

into amount, v_date from TaxPaid

where CitizenID = 101;

(3)

Select sum(AmountPaid) into total
from TaxPaid

where CitizenID = 101 and

DatePaid is between (*Date "01-01-2010" and
"31-12-2014") ;

If total > 10000 then status := "VIP";

~~else~~
elsif total is between 10000 and 8000 then
status := "BUSINESS";

else status := "NORMAL";

End if;

dbms_output.put_line ("Status:" || status);

End;

/.

(4)

Ans to the ques no: 02 (a)

A table with holidays are required.

Holidays (date-of-holiday, id (pk))

Leave_info (e_id, start, end, w-days)

Create or f replace function get_w-days (start In date,
end In date)

Return numbers

Is v_holidays numbers;

difference numbers;

no_of_week_numbers (3,0);

no_of_weekend_numbers;

total numbers;

Begin

Select date_of_holiday from count (date_of_holiday)

into v_holiday From Holidays

where date_of_holiday is between (start and
end);

difference := datediff (day, start, end);

(5)

no-of-week := difference/7 ;

no-of-weekend := no-of-week * 2 ;

total := difference - (no-of-weekend + v.holiday) ;

return total;

End;

/

Ans to the ques no: 02 (b)

A data cube allows data to be modeled and viewed in multiple dimensions. In data warehousing research architecture, datacube is representation of data cube is called a cuboid. Given a set of dimensions, we can generate of a cuboid. This cuboid show data in its different levels.

This levels are the no. of dimensions. For example for 3D cuboid, base will have time, location and item-name.

(6)

Ans to the ques no: 2 (c)

Data Warehousing:

Data warehousing is a subject-oriented, integrated, time variant and non-volatile collection of data in support of management's decision making process.

The key components of data warehousing are:-

(1) Subject-oriented:

It means that it focuses and provides only relevant information only skipping the ones which are not needed.

For example: Only sales per month is need not the transactional details.

(2) Integrated:

It is integrated & constructed from multiple sources like RDB, views, flat files and others.

For example: An information is taken from multiple relational data store.

(3)

(3) Time Variant:

Data stored to provide information from historic perspective i.e. the history.

for example: Data of last 5 years.

(4) Non-Volatile:

It is physically separate from operational environment and so it does not need any transaction processing. Only two ops operations are done: ~~to~~ Initial loading and then accessing.

For example: A data cannot be update here.

(8)

Ans to the que no: 02(c)

OLAP operations :-

(i) Roll-up:

performs aggregation by climbing up a hierarchy
or down by reduction.

For example :

Select region, country, sum(Sales)

From Sales

Group by Rollup(region, country);

(ii) Drill-down:

It is the opposite of roll-up's, that is, moving down
in hierarchy or creating new dimension.

For example :

Select region, sum(Sales)

From Sales

Group by Drilldown(region);

(3)

(3) Slice and Dice:

Slice causes selection of one dimension and dice causes selection of two or more dimensions.

For example :

Dice : Select city From Sales
Group by (Location, time);

Slice : Select city From Sales
Groupby slice(Location);

(4) Pivot :

It is the visualization that rotates data axes and provide an alternate presentation.

Ans to the ques no: 3(a)

Big Data is misnomer. ~~Because~~ The three properties of Big data are :-

(1) Volume : gives a huge volume of data

(2) Variety : Structured, unstructured, semi-structured data are included here

(3) Velocity : Data is accessed very fast. Here, The speed at which it is accessed is called stream computing.

(45)

Ans to the ques no: 3(b)

Because big data are large in volume and are predictive. So with time, use of data is increasing. So we use big data which are predictive and easier to query and load.

Q31 (11)

Ans to the ques no: 03(c)

Impediment Mismatch is the difference between Relational data store and in-memory datastore.

The in-memory is capable of storing dynamic and rich-complex structures. This is absent in RDBMS. As a result, there needs two translations in order to make storage. This creates impedance mismatch.

Ans to the ques no: 3(d)

Aggregate NoSQL database:

It is where there are aggregates which are collection of related objects; together represented as a single unit.

Hence complex structures are showed as a single unit. They are useful because:-

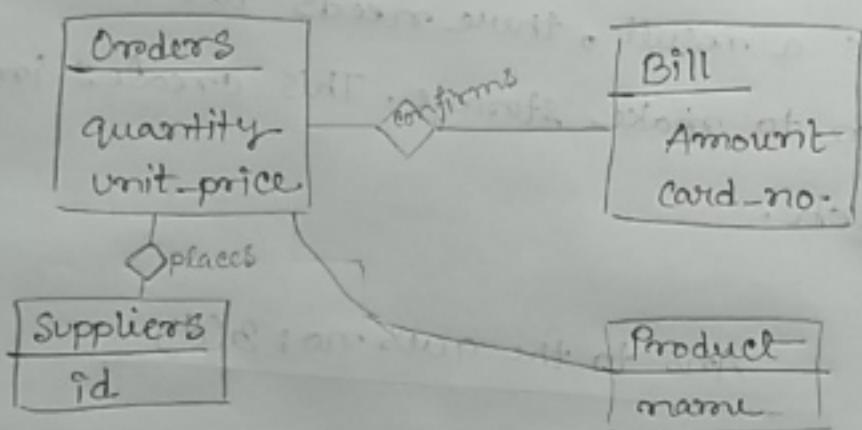
(1) They are non-relational and cluster friendly.

(2)

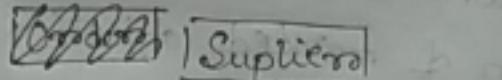
(2) Faster aggregation ~~and~~ answering queries such as sum, count and others.

(3) Faster loading.

In RD model :



In aggregate Model:



(13)

Main advantages over relational model:

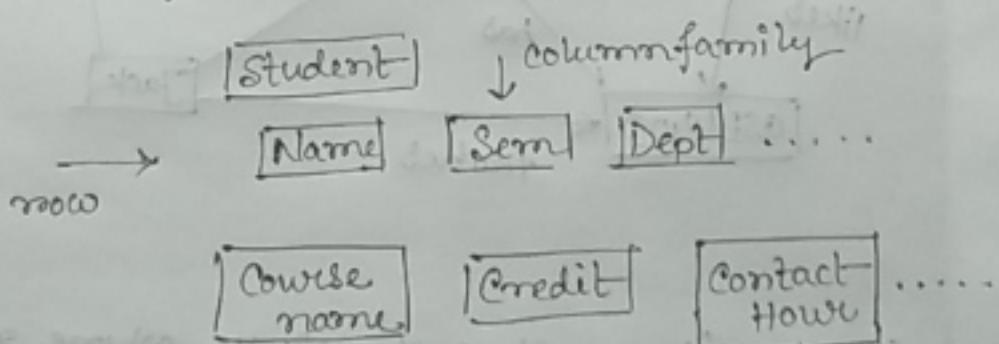
- (1) Efficient in partitioning
- (2) Fast queries
- (3) Cluster friendly
- (4) Scalable
- (5) Massive parallel processing is possible.

Ans to the que no: 03(e)

Column family Store:

Example:

In case of a university:



(Q4)

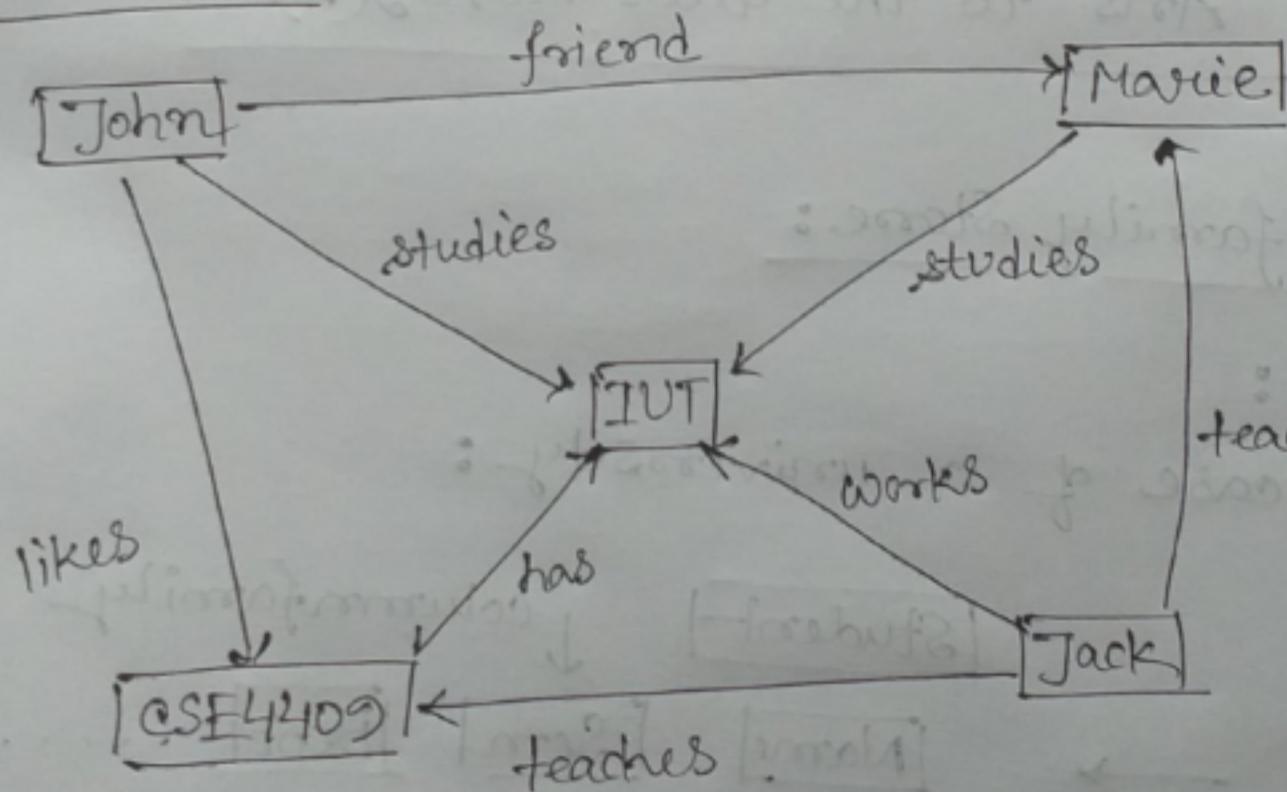
So far that:

"X"	180042101	CSE	...
Y	180042102	CSE	SWE

Ans to the ques no: 03(f)

Graph Database:

Example:



Graph is a ~~state~~ structure where much details are shown of each node and relations

(15)

between those nodes are also shown. It is ideal for capturing complex relationship such as for social media.

Ans to the ques no: 1(b)

Proposed solution will have a easy update of the system. ~~be without~~

```
create Index citizen_name_index  
On Citizen(Citizen_Name);
```