**Part 2 - Section 1:**

a)

INSERT INTO CustomerTBL (C\_ID, Address, Name, Age, Sum\_to\_date)

VALUES (123, ‘London-UK’, ‘Ali’, 24, 1000);

b)

SELECT c.Name, t.T\_ID

FROM CustomerTBL c, TicketTBL t, SalesTBL s

WHERE c.C\_ID = s.Customer.C\_ID

AND t.T\_ID = s.Ticket.T\_ID

AND s.S\_Date = ’04-05-2023’;

c)

SELECT \*

FROM FlexiRangeTBL f, IncludesTBL i

WHERE f.R\_ID = i.Range.R\_ID

AND i.Rate > 0.05;

d)

INSERT INTO IncludesTBL (Discount, Range, Rate)

Values (

(

SELECT REF(ft) FROM FlexibleTBL ft

WHERE ft.D\_ID = 111 AND ft.Date\_created = ’09-Apr-21’

),

(

SELECT REF(fr) FROM FlexiRangeTBL fr

WHERE fr.R\_ID = 333 AND fr.RangeStart = 0 AND fr.RangeEnd = 1000

),

0.1);

**Part 2 - Section 2:**

a) There are four combinations based on the 2 properties when (Update propagation) and where (Update location):

– Eager, primary copy

– Eager, update anywhere

– Lazy, primary copy

– Lazy, update anywhere

Eager Primary Copy:

In this scheme, updates take place at primary copy only. The primary eagerly propagates them to each secondary copy. A secondary copy is only allowed to process read-only transactions. All sites need to run a voting phase to ensure atomicity.

The primary site can propagate either:

* Update by update
* Wait until transaction has executed all operations, extract write set, and propagate all modifications in one message to each secondary copy.

To maintain transactions execution order at secondaries: messages broadcast in a FIFO order, according to the commit order at the primary. Primary initiates voting phase only once the previous has finished.

Recovery of the primary is challenging in this scheme.

b)

Checkpoints helps to reduce the time needed for recovery after failure. Checkpoint capture useful information regarding the running transactions and create timestamps in the log file that can help faster recovery after failure.

Database saves checkpoints at certain stages to avoid:

* Processing the entire log file in case of failure (especially if systems has been running for a long time this can be very time-consuming)
* To enable faster redoing and undoing of the operations after failures

It works by writing all log records in the main memory to secondary storage. Also, it writes the modified blocks in the database buffers to secondary storage.