**Oppgave 1.**

1. Secondary index - the index is defined on a non-ordering field of the data file.
2. Database recovery - Is the process of restoring the database to a correct state in the event of a failure.
3. XML and JSON databases are - document databases.
4. Software as a service - is software and data hosted on cloud. Accessed through using thin client interface (e.g. web browser). Consumer may be offered limited user specific application configuration services.
5. Atomicity - means that a transaction cannot be partially executed; it is either performed in its entirety or not performed at all.
6. Discretionary Access Control (DAC) - are applied through GRANT and REVOKE commands.
7. Deferred update - is the technique of writing updates to the database after a transaction has reached its commit point.
8. Triggers - are executed before, after, or instead of an SQL statement.
9. Two-tier client server architecture - consists of client process that requests resources and a server that provides resources.

**Oppgave 2.**

Et bilde som inneholder bord

Automatisk generert beskrivelse

* Create an XML document with all the necessary elements (including the root element) and tags using at least 2 rows of data:

<?xml version=”1.0” encoding=”UTF-8”?>

<DreamHouse>

<Branch>

<BranchNo> B005 </BranchNo>

<Street> 22 Deer Rd </Street>

<City> London </City>

<Postcode> SW1 4EH </Postcode>

</Branch>

<Branch>

<BranchNo> B003 </BranchNo>

<Street> 16 Argyl St </Street>

<City> Aberdeen </City>

<Postcode> AB2 3SU </Postcode>

</Branch>

</DreamHouse>

* Why is semi-structured data also called schema-less or self-describing? / What is semi-structured data?
* Because the data may have some structure, but the structure may not be rigid, regular or complete.
* The data is selfdescribing
* There’s no predefined schema
* The columns aren’t defined
* What are the five Vs of Big data?
* Volume (data size)
* Value (for the business)
* Velocity (refers to the speed at which new data is generating)
* Veracity (uncertainty of data)
* Variety (different forms of data; pictures, videos, sounds etc)

**Oppgave 3.**

Et bilde som inneholder tekst

Automatisk generert beskrivelse

* Write a trigger consisting on “A booking cannot be for a hotel room that is already booked for any of the specified dates”:

CREATE TRIGGER bookedAlready

BEFORE INSERT ON Booking

REFERENCING

OLD.RomNr = NEW.RomNr AND

OLD.DatoTil < NEW.DatoFra  
FOR EACH ROW

* What are the purposes of triggers? Use examples when answearing:

Triggers can be used to:

* Maintain the integrity of the data in the database.
  + For example, when a new record (representing a new employee) is added to the employees table (**INSERT event**), new records should also be created in the tables Taxes, Vacations, and Salaries.
* Move monitoring logic from applications to DBMSs, such as checking updates in the database.
  + For example, when a salary of an employee is **updated** from 1500 to 2000, the trigger fires on that event and checks that the update has actually happened.
* Log updates made on database tables.
* How is a stored procedure different from a function?
* A function will always return a value, while a stored procedure necessarily doesn’t have to return a value.

**Oppgave 5.**

* For each of the following schedules, state whether the schedule is conflict serializable or not conflict serializable. Justify your answer.

1) read(T1, balx), read(T2, balx), write(T1, balx), write(T2, balx), commit(T1), commit(T2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **T1** | R(x) |  | W(x) |  | commit |
| **T2** |  | R(x) |  | W(x) | commit |
| **T3** |  |  |  |  |  |

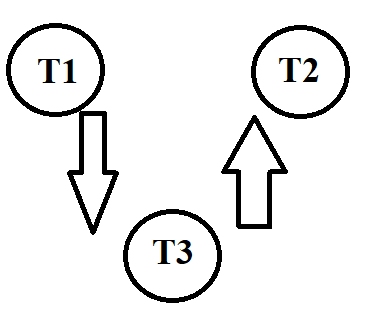
Et bilde som inneholder pil

Automatisk generert beskrivelse

* Theres a conflict from T1 R(x) to T2 W(x) and also a conflict from T2 (Rx) to T1 W(x), hence there’s a cycle and the schedule is not conflict serializable.

2) read(T1, balx), read(T2, baly), write(T3, balx), read(T2, balx), read(T1, baly), commit(T1), commit(T2)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **T1** | R(x) |  |  |  | R(y) | Commit |
| **T2** |  | R(y) |  | R(x) |  | commit |
| **T3** |  |  | W(x) |  |  |  |



* There’s no cycle in the schedule, hence this schedule is conflict serializable. The schedule after conflict serialization is as follows:
  + T1: R(x); R(y)

T2: R(y); R(x)

T3: W(x)

* Describe, with examples, three types of problem that can occur in a multi-user environment when concurrent access to the database is allowed.
* Lost update, which means that two transactions are trying to overwrite the same variables. This can be prevented by avoiding reading the variable from the second transaction until the first transaction is completed.
* Dirty read, which means that one transaction is just checking a function but not committing (it uses rollback) while the second transaction is operating with inconsistent (dirty) variables. Can be prevented by avoiding reading the variable for one transaction until the second transaction is complete.
* Inconsistent analyzes, which mean that one transaction is reading the variables while another transaction is overwriting the same variables. Can be prevented by avoiding reading the variables for one transaction until the second transaction is completed.
* Explain a concurrency problem technique to solve the problems you described in previous question.
* **I just did**
* What are deadlocks and how could you solve them?
* A deadlock is a situation where two transactions are dependent on the same variables which are locked. For example, when T1 is reading x and writing y; and T2 is reading y and writing x. T1 is locking the variables for T2 and vice versa.