**Unit 11 Reading**

Sarkar, T. & Roychowdhury, S. (2019) Data Wrangling with Python. 1st ed. Packt.

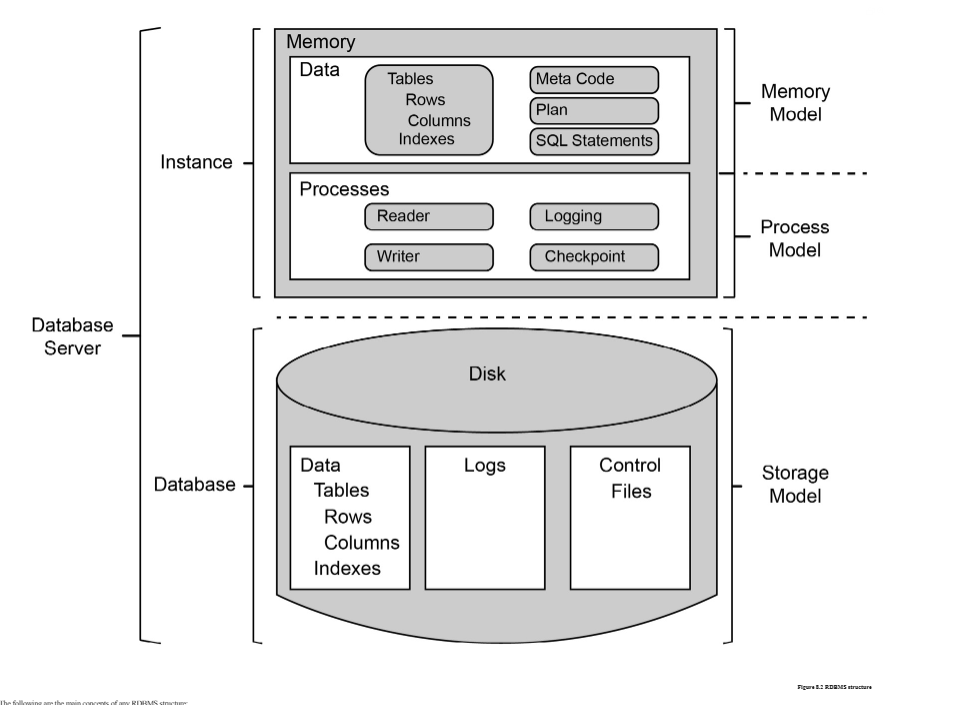
**Chapter 8 – RDMS and SQL**

RDMS (Relational database management system) are well used because:

* Safe way to store, manage and retrieve data
* Solid foundation in maths (relational algebra and calculus) plus SQL efficient and intuitive for interaction.
* Well tested languages, ell understood – tried and tested
* Scaling RDMS well understood and many experienced data base admins to do it.

RDMS is software that manages data (in tabular form for end user) built using Codd’s relational model. Shift in recently to NoSQL (MongoDB for example) – these in some aspect follow RDMS rules and in many areas reject them.

Three main elements – storage engine, query engine and log management:



Storage engine - stores data in efficient way and gives it back when requested. End user including developer will not access this directly.

Query engine – allows creation of data objects such as tables, manipulate them (add columns, add rows) and query (read rows) using SQL

Log management – creates and maintains logs, handles replication and partitions.

SQL (Structured Query Language) – domain specific language used in databases to define, insert, manipulate and retrieve data. It can be sub-divided into four smaller sub-languages, namely DDL (Data Definition Language), DML (Data Manipulation Language), DQL (Data Query Language), and DCL (Data Control Language).

* Easy to understand
* Out of box way to manipulate data
* Fast readable way to work with data
* Declarative language – what to be done, not how to do it. So can focus on query problems.

DDL – defines data structure e.g CREATE TABLE

DML – modify actual data such as insert, update e.g INSERT INTO

DQL – query data in system – e.g SELECT

Once data and tables created shows as a table with columns. These have keys which are unique for each row, unique values can be elsewhere but primary key identify and cannot be null. Primary key in one table and foreign key in another can join data tables together by relationships.

**Activities on using a RDBMS data with a connection from Python**

Activities use SQLite, this is transferable to other databases such as Oracle, MySQL, Postgresql, and DB2.

“Most of the industry standard projects which are written in Python and use some kind of RDBMS as the data store, most often relay on an ORM or Object Relational Mapper. An ORM is a high-level library in Python which makes many tasks, while dealing with RDBMS, easier. It also exposes a more Pythonic API than writing raw SQL inside Python code.”

After connecting, exercise shows how to create tables, specific data types and primary keys and commit the changes. This is the DDL (Data definition language).

The SELECT clause can be used in python SQLite to select rows with different SQL criteria.

ALTER TABLE and ADD COLUMN to update the table further.

**Relation Mapping in databases**

In an RDMS the power comes from relationships between entities and being able to manage the relationships of data. When two tables and specific primary and then foreign keys, it’s possible to tell RDMS what to do with linked records. So for example delete the children of the main record in linked tables when the main record is deleted.

Example is given of a user table having one user but a linked comments table with many comments per user, a foreign key of the user ID is in the comments table. It is possible to tell the RDMS to delete the child comment records in the linked table. ON DELETE CASCADE is used in the example.

Exercise continues demonstrating joins using primary key, foreign key relationship from one table to get data from others linked via relationships.

Then exercises on how to connect to a database and answer analysis questions directly in Python

**Reflections**

A refresher on SQL and seeing the flexibility of both creating databases and querying them directly using SQL within Python code demonstrates a high level of flexibility to be able to wrangle, create, manage and query data.