CINS 370 Introduction to Databases
Homework 1

1.1 Define the following terms:

Data: Data can be defined as a representation of facts, concepts, or instructions in a formalized manner, which should be suitable for communication, interpretation, or processing by human or electronic machine.

Database: A collection of data arranged for ease and speed of search and retrieval.

DBMS: A database management system (DBMS) is a software package designed to define, manipulate, retrieve and manage data in a database. A DBMS generally manipulates the data itself, the data format, field names, record structure and file structure.

Database system: A database system is a high-level definition of the structure and relationship between stored data, a database or databases, users and the hardware or operating system used for the storage.

Database catalog: The database catalog of a database instance consists of metadata in which definitions of database objects such as base tables, views (virtual tables), synonyms, value ranges, indexes, users, and user groups are stored.

Program- data independence: Program-data independence refers to the capability of leaving data intact and accessible regardless of modifications to the database that contains the data and this independence allows database administrators to retool a database to meet an enterprise's new information needs without worrying that the people who need the data for research, reports and making informed decisions will suddenly be unable to access it.

User view: A view of part or all of the contents of a database specified to facilitate a particular purpose or user activity. It is a partial and/or redefined description of the logical schema of the database.

DBA: Database administration is the function of managing and maintaining database management systems (DBMS) software.

End user: The ultimate consumer of a product for whom the product has been designed.

Canned transaction: Canned transactions are standard types of queries and updates which frequently used by Naive or parametric end users to constantly querying and updating database

Deductive database system: A deductive database is an advanced database system that can make deductions (i.e., conclude additional facts) based on rules and facts stored in the database. It uses an inference system that, by evaluating rules against facts, derives new facts which in turn can be used to answer queries.

Persistent object: Persistent data is stored outside of a transaction context, and so survives transaction updates. Usually the term persistent data is used to indicate the databases that are shared, accessed and updated across transactions. The two main strategies used to create and identify persistent objects are: Persistence extensions and Persistence through reachability

Metadata: Metadata describes other data. It provides information about a certain item's content.

Transaction-processing application: Transaction processing is the process of completing a task and/or user/program request either instantly or at runtime.

1.4 What are the responsibilities of the DBA and the database engineers?
D.B.A. refers to database administrator. He is mainly responsible for settings and management tasks related to a database. The maintenance, optimization, recovery, etc., are included in the profile of a Database Administrator. Generally, a Database Engineer also works as a Database Administrator or supervises a Database Administrator.

1.6 Discuss the capabilities that should be provided by a DBMS.

- **Data Storage, Retrieval & Update:** The DBMS software allows the app to read data from the database, add new data into the database, modify the data or delete the data from the database.

- **Master Catalog**: The DBMS software stores all the info it needs about itself within itself.

- **Central Control, Monitoring**: The DBMS software includes features that allow database administrators (DBAs) to watch the database software as it runs to make sure it is running smoothly and efficiently, and that the database remains available and reliable.

- **Transaction Support:** When an application or user needs to process a transaction that updates the database, the DBMS software must keep track of ALL aspects of that transaction and ensure that the entire transaction is completed successfully. If any part of the transaction fails, then the DBMS software must be able to "roll back" any partial updates, leaving the database in a consistent state.

- **Data Independence ("Views"):** In cases where the different kinds or types of data stored in a database may not be modified or viewed by all users, the DBMS software allows DBAs to create "views" of subsets of data that are appropriate for the permissions granted to each user.

- **Concurrent Shared Updates & Locking:** In cases where there are multiple users accessing the database at the same time, the DBMS software must control access so that one user's updates don't interfere with another user's access. The DBMS implements this control via "locks". If one user is updating a record, that record is locked so that no one else can read it or update it until the first user's update is finished. If one user is reading a record, it is locked so that no one can update it until the read activity is finished. Example is Google docs.

- **Integrity Constraints:** Data is stored in relational databases in TABLES. Often, the data in one table is related to the data in another table. For example, the database may store customer data in one table, and a history of each purchase transaction for that customer in another table. The DBMS software can ensure that every purchase transaction ties back to a valid customer and, for example, prevent the insertion of a purchase transaction for a customer that is not on file. This is an example of the DBMS software enforcing a data integrity constraint.

- **Query Language.** Most DBMS software products come with a query language that allows users of the DBMS software to write programs that access the data in the database. "SQL" (Structured Query Language) is a very common programming tool used by relational databases to enable users to work with their databases.

1.12 Cite some examples of integrity constraints that you think can apply to the database.

Few constraints that can be imposed database are:

- Grade can be given only to enrolled students.
- Each section must belong to any course.
- **Key-constraints** applies to **Grade_report** as student number have been repeated.

- Prerequisite of each course must have been offered course in part or must be an existing course.
- **Referential Integrity Constraints** applies to student-grade report and section-grade report.
- Each course must be a part of existing department.
- Students must be part of section for which it is graded.

2.1 Define the following terms:

Data model: A data model refers to the logical inter-relationships and data flow between different data elements involved in the information world. It also documents the way data is stored and retrieved.

Database schema: A database schema represents the logical configuration of all or part of a relational database. It can exist both as a visual representation and as a set of formulas known as integrity constraints that govern a database. These formulas are expressed in a data definition language, such as SQL.

Database state: A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

Internal schema: The internal schema defines the physical storage structure of the database. The internal schema is a very low-level representation of the entire database. It contains multiple occurrences of multiple types of internal record. In the ANSI term, it is also called "stored record'.

Conceptual schema: The conceptual schema describes the Database structure of the whole database for the community of users. This schema hides information about the physical storage structures and focuses on describing data types, entities, relationships, etc. This logical level comes between the user level and physical storage view.

External schema: An external schema describes the part of the database which specific user is interested in. It hides the unrelated details of the database from the user. There may be "n" number of external views for each database

Data independence: Data independence is the idea that generated and stored data should be kept separate from applications that use the data for computing and presentation.

DDL: DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.

DML: DML is short name of Data Manipulation Language which deals with data manipulation and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE, etc., and it is used to store, modify, retrieve, delete and update data in a database.

SDL (Storage definition language): It is used to specify the internal schema. The mappings between the two schemas may be specified in either one of these languages. In most relational DBMSs today, there is no specific language that performs the role of SDL.

VDL (**View Definition Language):** This language is used to specify user views and their mapping to conceptual schema. It defines the subset of records available to classes of users. It creates virtual tables and the view appears to users like conceptual level. It specifies user interfaces.

Query Language: Structured Query Language is a standard Database language which is used to create, maintain and retrieve the relational database.

Host language: Host Language is the computer language which either high level are low level data manipulation commands embedded in general purpose programming language.

Data sublanguage: It is a portion of a language that is required to manage and modify
A relational DBMS.

Database utility: The database utility is the interface between the ABAP Dictionary and the relational database underlying the SAP system. The database utility allows you to edit (create, delete and adjust to changes to their definition in the ABAP Dictionary) database objects derived from objects of the ABAP Dictionary.

Catalog: Catalog complete description of the database structure and constraints.

Client/server architecture: Server Architecture database architecture which system functionality: client model (application), server model (landless storage)

Three-tier architecture: The 3-Tier architecture contains another layer between the client and server. In this architecture, client can't directly communicate with the server. The application on the client-end interacts with an application server which further communicates with the database system.

n-tier architecture: The nth tier only has to know how to handle a request from the h+1 tier and forward to h-1 tier.

**2.3 What is the difference between a database schema and a database state?
this architecture**

A database schema or an intention gives a description of the database. This can be considered as a blueprint of a database and gives a list of fields in the database with their data types. In other words, it describes the organization and structure of data in a database system, along with the relationships mapped between the entities.

A database state provides the present state of the database and its data. It can be considered as an extension of the database schema. When a database is newly defined, the corresponding database state is empty. Factors that affect a database state are entering, deleting or modifying information in the database.

2.4 Describe the three-schema architecture. Why do we need mappings among schema levels? How do different schema definition languages support?

- The three schema architecture is also called ANSI/SPARC architecture or three-level architecture.
- This framework is used to describe the structure of a specific database system.
- The three schema architecture is also used to separate the user applications and physical database.
- The three schema architecture contains three-levels. It breaks the database down into three different categories.

  Internal Level
  The internal level has an internal schema which describes the physical storage structure of the database.

  Conceptual Level The conceptual schema describes the design of a database at the conceptual level. Conceptual level is also known as logical level.

External Level At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database.

b) The need for mappings between schema levels appears due to visualization and schema matching. This mapping helps in different types of transformation. A Database Management System has three schema levels; Physical or internal schema, Conceptual or logical schema and External or view level schema.

c) In a three-schema approach, most data-related description languages or tools associated with schemas focus on the "physical level" and "view level", with the "conceptual level" mostly used in combining the schema design itself. In relational databases, the physical model is explained using SQL DDL. Physical schemas in NoSQL can be fixed to particular records using JSON or XML. There are some languages to describe conceptual schemas. But nowadays languages are not used for conceptual schemas but instead Entity relationship model, a diagramming and description tool is used. The "view level" or "external level" is implemented outside of data managers in user-side code. This is done using object relational mapping with the help of tools like Hibernate. These tools convert "physical layer" schemas structure to "external-friendly" structures. Here the user input UI elements is given to the application code. Code having "database layer" is implemented inside the API.

DATASET RETRIEVAL (45 points):

Dataset 1: Car-Emission

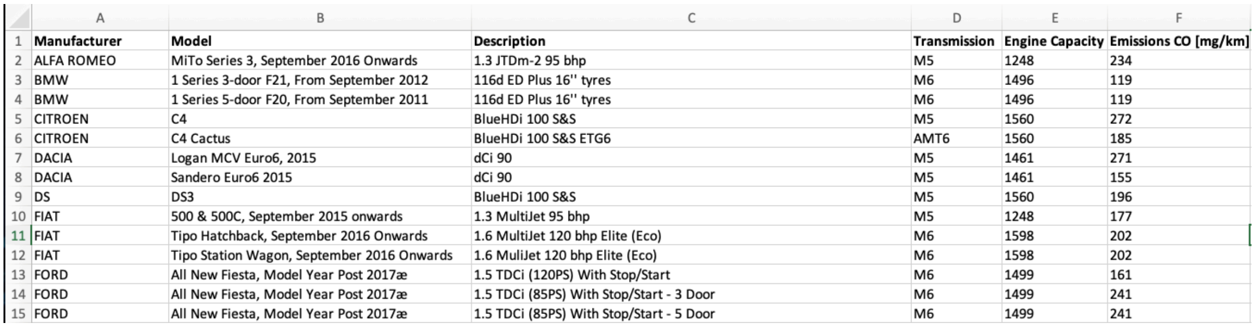
Number of rows: 5119

Number of columns: 6

Reference: https://www.kaggle.com/dananos/car-emissions-data

Attributes:

- *Manufacturer*: Company of the car
- *Model*: Model of that Company
- *Description*: Variant of that Model
- *Transmission*: Transmission of M type
- *Engine Capacity*: Overall engine capacity
- *Emission*: Emission of that model.



	A	B	C	D	E	F
1	**Manufacturer**	**Model**	**Description**	**Transmission**	**Engine Capacity**	**Emissions CO [mg/km]**
2	ALFA ROMEO	MiTo Series 3, September 2016 Onwards	1.3 JTDm-2 95 bhp	M5	1248	234
3	BMW	1 Series 3-door F21, From September 2012	116d ED Plus 16'' tyres	M6	1496	119
4	BMW	1 Series 5-door F20, From September 2011	116d ED Plus 16'' tyres	M6	1496	119
5	CITROEN	C4	BlueHDi 100 S&S	M5	1560	272
6	CITROEN	C4 Cactus	BlueHDi 100 S&S ETG6	AMT6	1560	185
7	DACIA	Logan MCV Euro6, 2015	dCi 90	M5	1461	271
8	DACIA	Sandero Euro6 2015	dCi 90	M5	1461	155
9	DS	DS3	BlueHDi 100 S&S	M5	1560	196
10	FIAT	500 & 500C, September 2015 onwards	1.3 MultiJet 95 bhp	M5	1248	177
11	FIAT	Tipo Hatchback, September 2016 Onwards	1.6 MultiJet 120 bhp Elite (Eco)	M6	1598	202
12	FIAT	Tipo Station Wagon, September 2016 Onwards	1.6 MuliJet 120 bhp Elite (Eco)	M6	1598	202
13	FORD	All New Fiesta, Model Year Post 2017æ	1.5 TDCi (120PS) With Stop/Start	M6	1499	161
14	FORD	All New Fiesta, Model Year Post 2017æ	1.5 TDCi (85PS) With Stop/Start - 3 Door	M6	1499	241
15	FORD	All New Fiesta, Model Year Post 2017æ	1.5 TDCi (85PS) With Stop/Start - 5 Door	M6	1499	241

Dataset 2: Car-Sales

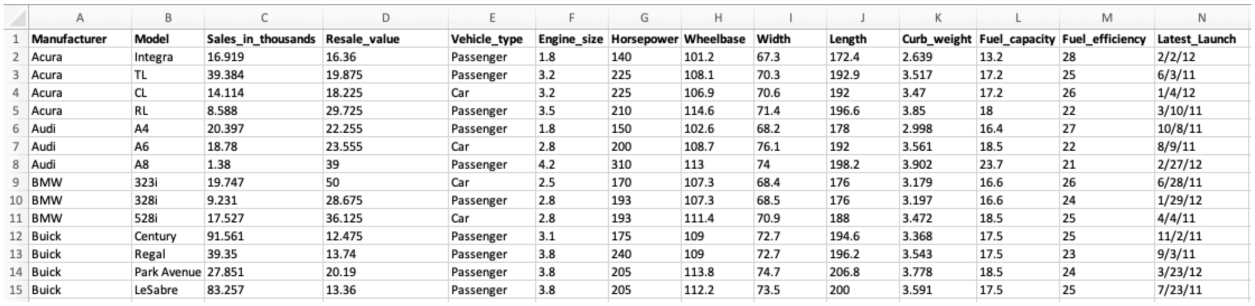
Number of rows: 158

Number of columns: 14

Reference: https://www.kaggle.com/gagandeep16/car-sales

Attributes:

- *Manufacturer*: Company of the car
- *Model*: Model of that Company
- *Sale*: Model sold in thousand
- *Resale value: Amount received after selling the car*
- *Engine Capacity*: Overall engine capacity
- *Horsepower: Engine Power measuring term*
- *Wheelbase: Ground clearance*
- *Width: Overall length of car*
- *Length: Overall length of car*
- *Weight: Car weight in tons*
- *Fuel capacity: Maximum capacity of fuel for that model*
- *Fuel efficiency:* Car travel in miles per 1 gallon
- *Launch:* Launch date of that model



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	**Manufacturer**	**Model**	**Sales_in_thousands**	**Resale_value**	**Vehicle_type**	**Engine_size**	**Horsepower**	**Wheelbase**	**Width**	**Length**	**Curb_weight**	**Fuel_capacity**	**Fuel_efficiency**	**Latest_Launch**
2	Acura	Integra	16.919	16.36	Passenger	1.8	140	101.2	67.3	172.4	2.639	13.2	28	2/2/12
3	Acura	TL	39.384	19.875	Passenger	3.2	225	108.1	70.3	192.9	3.517	17.2	25	6/3/11
4	Acura	CL	14.114	18.225	Car	3.2	225	106.9	70.6	192	3.47	17.2	26	1/4/12
5	Acura	RL	8.588	29.725	Passenger	3.5	210	114.6	71.4	196.6	3.85	18	22	3/10/11
6	Audi	A4	20.397	22.255	Passenger	1.8	150	102.6	68.2	178	2.998	16.4	27	10/8/11
7	Audi	A6	18.78	23.555	Car	2.8	200	108.7	76.1	192	3.561	18.5	22	8/9/11
8	Audi	A8	1.38	39	Passenger	4.2	310	113	74	198.2	3.902	23.7	21	2/27/12
9	BMW	323i	19.747	50	Car	2.5	170	107.3	68.4	176	3.179	16.6	26	6/28/11
10	BMW	328i	9.231	28.675	Passenger	2.8	193	107.3	68.5	176	3.197	16.6	24	1/29/12
11	BMW	528i	17.527	36.125	Car	2.8	193	111.4	70.9	188	3.472	18.5	25	4/4/11
12	Buick	Century	91.561	12.475	Passenger	3.1	175	109	72.7	194.6	3.368	17.5	25	11/2/11
13	Buick	Regal	39.35	13.74	Passenger	3.8	240	109	72.7	196.2	3.543	17.5	23	9/3/11
14	Buick	Park Avenue	27.851	20.19	Passenger	3.8	205	113.8	74.7	206.8	3.778	18.5	24	3/23/12
15	Buick	LeSabre	83.257	13.36	Passenger	3.8	205	112.2	73.5	200	3.591	17.5	25	7/23/11

Dataset 3: Cricket-Players

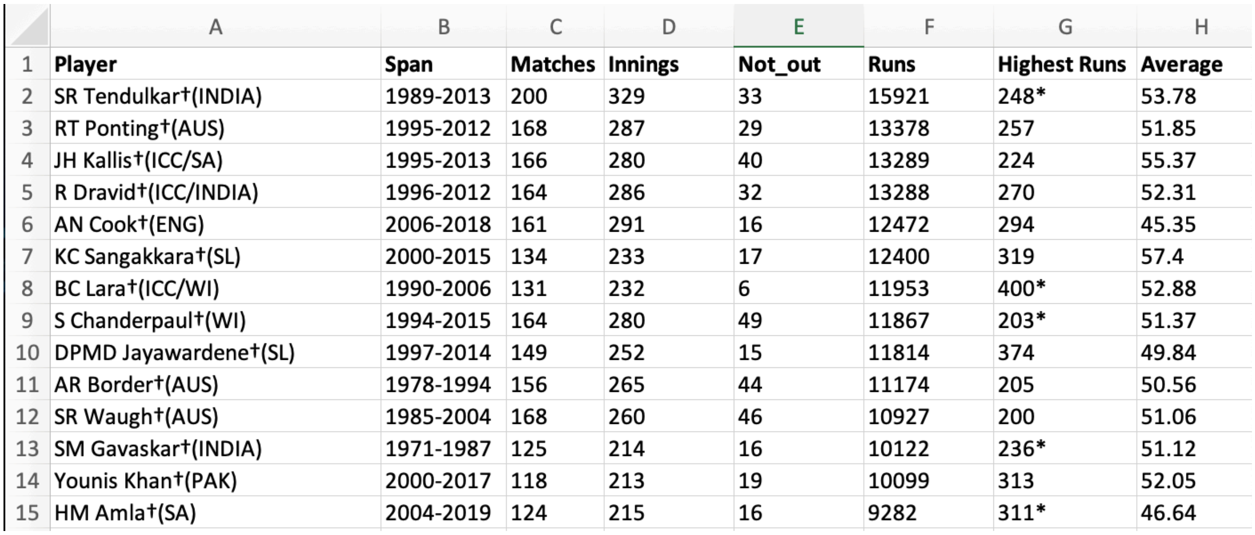
Number of rows: 3002

Number of columns: 8

Reference: https://www.kaggle.com/veeralakrishna/icc-test-cricket-runs

Attributes:

- *Player:* Player name
- *Span:* Time for which player played cricket
- *Matches: Number of matches played by the player*
- *Innings: Number of Innings played by the player*
- *Not out: Number of the player was not-out in the innings*
- *Runs: Number of runs scored by the player*
- *Highest runs: Highest runs scored by player in one inning or matches*
- *Average: Batting average of that player*



	A	B	C	D	E	F	G	H
1	**Player**	**Span**	**Matches**	**Innings**	**Not_out**	**Runs**	**Highest Runs**	**Average**
2	SR Tendulkar†(INDIA)	1989-2013	200	329	33	15921	248*	53.78
3	RT Ponting†(AUS)	1995-2012	168	287	29	13378	257	51.85
4	JH Kallis†(ICC/SA)	1995-2013	166	280	40	13289	224	55.37
5	R Dravid†(ICC/INDIA)	1996-2012	164	286	32	13288	270	52.31
6	AN Cook†(ENG)	2006-2018	161	291	16	12472	294	45.35
7	KC Sangakkara†(SL)	2000-2015	134	233	17	12400	319	57.4
8	BC Lara†(ICC/WI)	1990-2006	131	232	6	11953	400*	52.88
9	S Chanderpaul†(WI)	1994-2015	164	280	49	11867	203*	51.37
10	DPMD Jayawardene†(SL)	1997-2014	149	252	15	11814	374	49.84
11	AR Border†(AUS)	1978-1994	156	265	44	11174	205	50.56
12	SR Waugh†(AUS)	1985-2004	168	260	46	10927	200	51.06
13	SM Gavaskar†(INDIA)	1971-1987	125	214	16	10122	236*	51.12
14	Younis Khan†(PAK)	2000-2017	118	213	19	10099	313	52.05
15	HM Amla†(SA)	2004-2019	124	215	16	9282	311*	46.64