Welcome to General Assembly

- **WiFi** GA Guest
- **Password** yellowpencil

GENERAL ASSEMBLY

DATA SCIENCE

Lesson 10 - Databases and SQL

Course Plan

UNITS

UNIT 1: FOUNDATIONS OF DATA MODELING	Introduction to Data Science	Lesson 1
	Elements of Data Science	Lesson 2
	Data Visualisation	Lesson 3
	Linear Regression	Lesson 4
	Logistic Regression	Lesson 5
	Model Evaluation	Lesson 6
	Regularisation	Lesson 7
	Clustering	Lesson 8
UNIT 2: DATA SCIENCE IN THE REAL WORLD	Recommendations	Lesson 9
	SQL + Productivity	Lesson 10
	Decision Trees	Lesson 11
	Ensembles	Lesson 12
	Natural Language Programming	Lesson 13
	Cloud Computing	Lesson 14
	Time Series	Lesson 15
	Soft Skills	Lesson 16
	Network Analysis	Lesson 17
	Neural Networks	Lesson 18
	Final Projects Presentations	Lesson 19
	Final Projects Presentations	Lesson 20

Paul & James review final project ideas

Git & GitHub – 1 Pager Guide!

(Part B) EVERY CLASS:

At the START of the class, you'll need to sync the latest materials from the COURSE repo:

(1) **Make sure you are in the dat11syd directory:**

```
cd ~/workspace/dat11syd
```

(2) **Make sure to select the "master" branch of your repo:**

```
git checkout master
```

(3) **Fetch the latest changes from the UPSTREAM repo (i.e the course repo)**

```
git fetch upstream
```

(4) **Merge the changes from the upstream repo to your master branch:**

```
git merge upstream/master
```

DURING the class:

(5) **Before editing, either copy files to your "students/" folder, or rename them**

At the END of every class:

(6) **Make sure you are in the dat11syd directory:**

```
cd ~/workspace/dat11syd
```

(7) **Add any files that you've updated to your git registry:**

```
git add -A
```

(8) **Commit the changes with a sensible comment:**

```
git commit -m "my updates for lesson 7"
```

(9) **Push your changes to your PERSONAL repo:**

```
git push origin master
```

DONE!!!!!

AGENDA 5

1. Databases
2. SQL
3. Lab: SQL
4. Discussion of Data Science Productivity Tools

DATA SCIENCE PART TIME COURSE

DATABASES


Microsoft
SQL Server
ORACLE
PostgreSQL
MySQL
SYBASE
IBM DB2

Databases are computer systems that manage storage and querying of data. Databases provide a way to organise data along with efficient methods to retrieve specific information.

Typically, retrieval is performed using a query language, a mini programming syntax with a few basic operators for data transformation, the most common of which is SQL.

Databases are the standard solution for data storage and are much more robust than text, CSV or json files. Most analyses involve pulling data to and from a resource and in most settings, that means using a database.

Databases can come in many flavours, designed to serve for different use cases.

WHY USE DATABASES

Rules on structure make writing and retrieving data more reliable and efficient.

- Standardised business definitions
- Source of truth
- Fast read

Where databases are valued:

- Operational systems
- Reporting

RELATIONAL DATABASE - TABLES

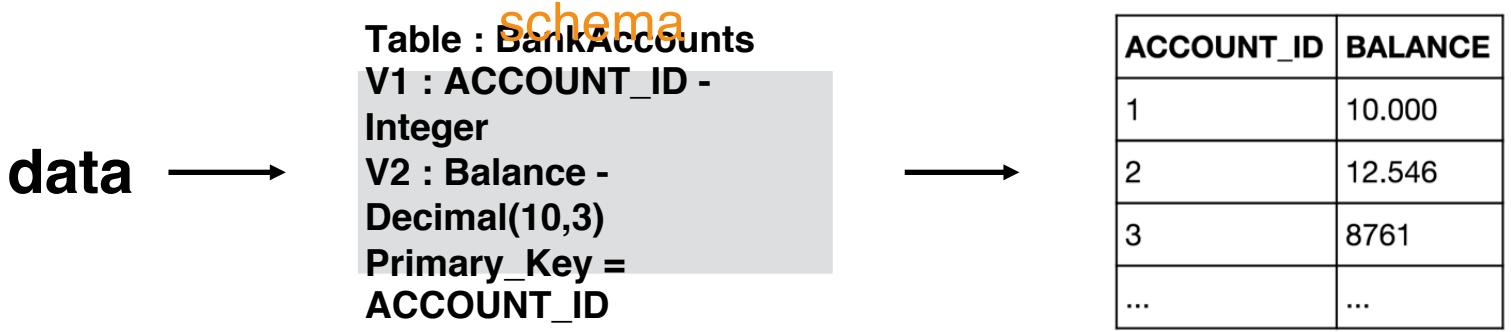
A relational database is a database based tabular data and links between data entities or concepts.

A relational database is organised into tables. Each table should correspond to one entity or concept.

ACCOUNT_ID	BALANCE
1	10.000
2	12.546
3	8761
...	...

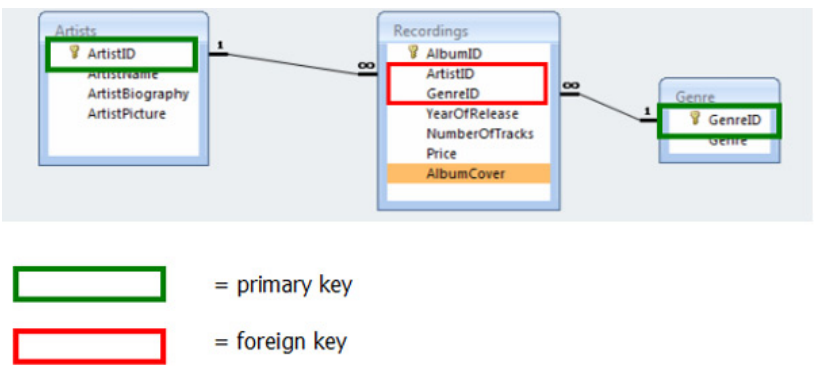
A table is made up rows and columns, similar to a Pandas dataframe or Excel spreadsheet.

A table also has a schema which is a set of rules for what goes in each table. These specify what columns are contained in the table and what type those columns are (text, integers, floats, etc.).



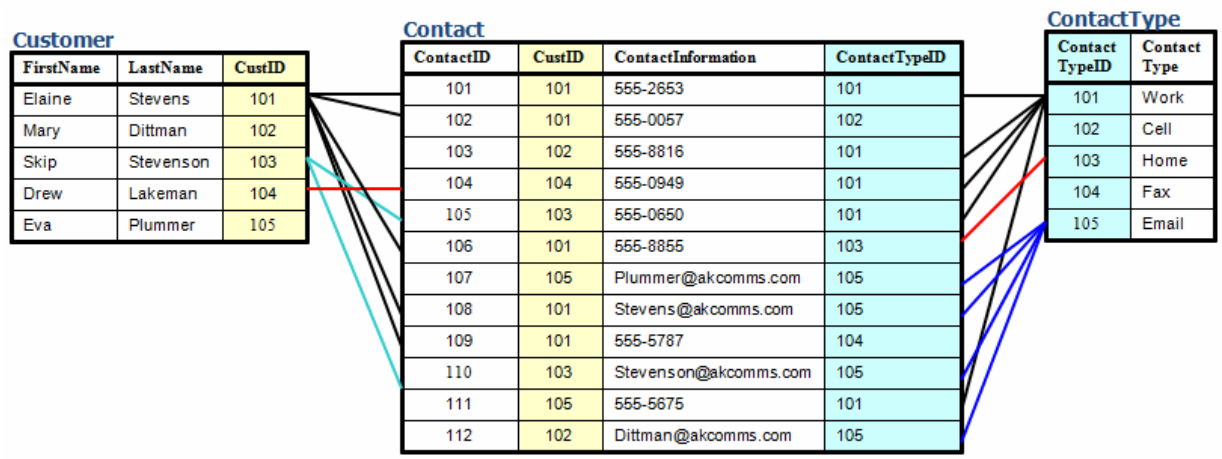
Each table typically has a primary key column. This column is a unique value per row and serves as the identifier for the row.

A table can have many foreign keys as well. A foreign key is a column that contains values to link the table to the other tables.



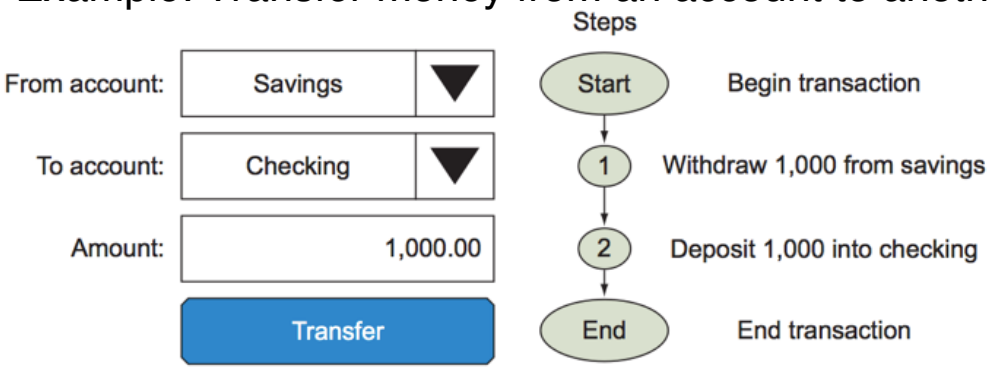
RELATIONAL DATABASE - DATA MODEL

Databases are ‘modelled’ to suit their intended purpose.



A unit of work performed against a database is called a transaction. This term generally represents any change in database.

Example: Transfer money from an account to another.



- What happens if step 1 succeeds and step 2 fails ?
- What if you request the balance between step 1 and step2 ?

ACID is a set of properties that guarantee that database transactions are processed reliably.

Atomicity "all or nothing": if one part of the transaction fails, the entire transaction fails, and the database state is left unchanged.

Consistency ensures that any transaction will bring the database from one valid state to another.

Isolation ensures that the concurrent execution of transactions results in a system state that would be obtained if transactions were executed serially, i.e., one after the other.

Durability ensures that once a transaction has been committed, it will remain so, even in the event of power loss, crashes, or errors.

DATA SCIENCE PART TIME COURSE

NO-SQL Databases

SQL

- Traditional rows and columns data
- Strict structure / Primary Keys
- Entire column for each feature
- Industry standard

NoSQL

- No well defined data structure
- Works better for unstructured data
- Cheaper hardware
- Popular among Startups

SQL

- MySQL
- Oracle
- Postgres
- SQLite
- SQLServer
- Redshift

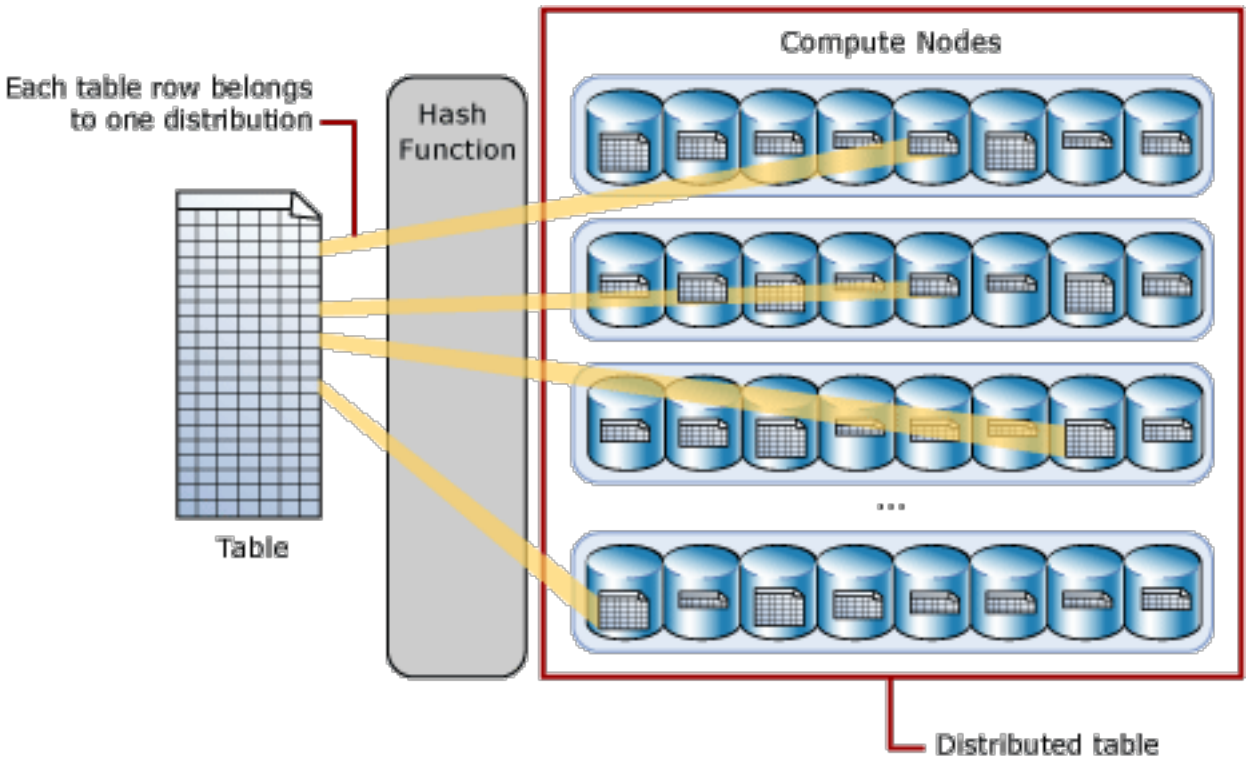
NoSQL

- MongoDB
- CouchDB
- Redis
- Cassandra
- Neo4j
- HBase

DATA SCIENCE PART TIME COURSE

BIG DATA & SQL

MPP RDBMS - Massively Parallel Processing



TERADATA

Greenplum

VERTICA
An HP Company

AMAZON REDSHIFT

1. What is it?
2. Sign up https://community.cloud.databricks.com
3. Import URL https://docs.databricks.com/_static/notebooks/gentle-introduction-to-apache-spark.html

DATA SCIENCE PART TIME COURSE

SQL

- http://rextester.com/l/sql_server_online_compiler
- How to make a table?
- How to query a table
- How to filter data (WHERE CLAUSE)
- How to Group by
- Aggregations (sum, average, count)
- Joins

SQL

Make Table:

```
Create table GA_SALES (
    month_name nvarchar(20)
    ,qtr nvarchar(20)
    ,sales_dollars money
);
```

```
-- Insert Rows:
insert into GA_SALES values ('Jul', 'FYQ1', 20000.5);
insert into GA_SALES values ('Aug', 'FYQ1', 30000);
insert into GA_SALES values ('Sep', 'FYQ1', 40000);
```

Query:

```
select top 10 * from GA_SALES; — (limit 10)
```

Where Clause:

```
select * from GA_SALES where sales_dollars > 25000;
```

What Tables?

```
select * from information_schema.tables
```

Group by:

```
select qtr
        ,sum(sales_dollars) as qtr_sales
from GA_SALES
group by qtr
```

Aggregations:

Try - avg() or count()

Joins:

```
create table GA_TARGETS (
    qtr nvarchar(20)
    ,target money
);

insert into GA_TARGETS values ('FYQ1', 100000);
insert into GA_TARGETS values ('FYQ2', 120000);

select a.qtr
   ,max(target) as target
   ,sum(sales_dollars) as qtr_sales
   ,max(target) - sum(sales_dollars) variance
from GA_SALES a
```

LAB

DATA SCIENCE PART TIME COURSE

DISCUSSION: PRODUCTIVITY TOOLS

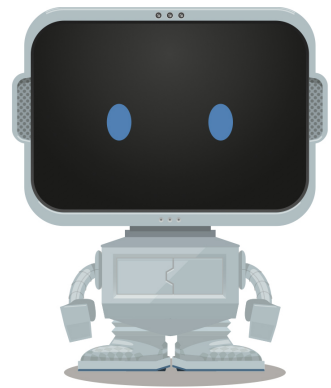
PRODUCTIVITY TOOLS

Workflow, collaboration, operationalising ⟷ Automatic Machine Learning - Auto ML

rapidminer

scienceOPS

sas



DataRobot

Auto-Sklearn

DATA SCIENCE

HOMEWORK

Read the following

Read Chapter 2 of http://www.redbook.io/

https://medium.com/airbnb-engineering/automated-machine-learning-a-paradigm-shift-that-accelerates-data-scientist-productivity-airbnb-f1f8a10d61f8