



DATA ENGINEER

COURSE OUTLINE

- Introduction to Data Engineering
- Data Modeling and Database Design
- Overview of ETL
- SQL RECAP
- ETL LAB(SQL)

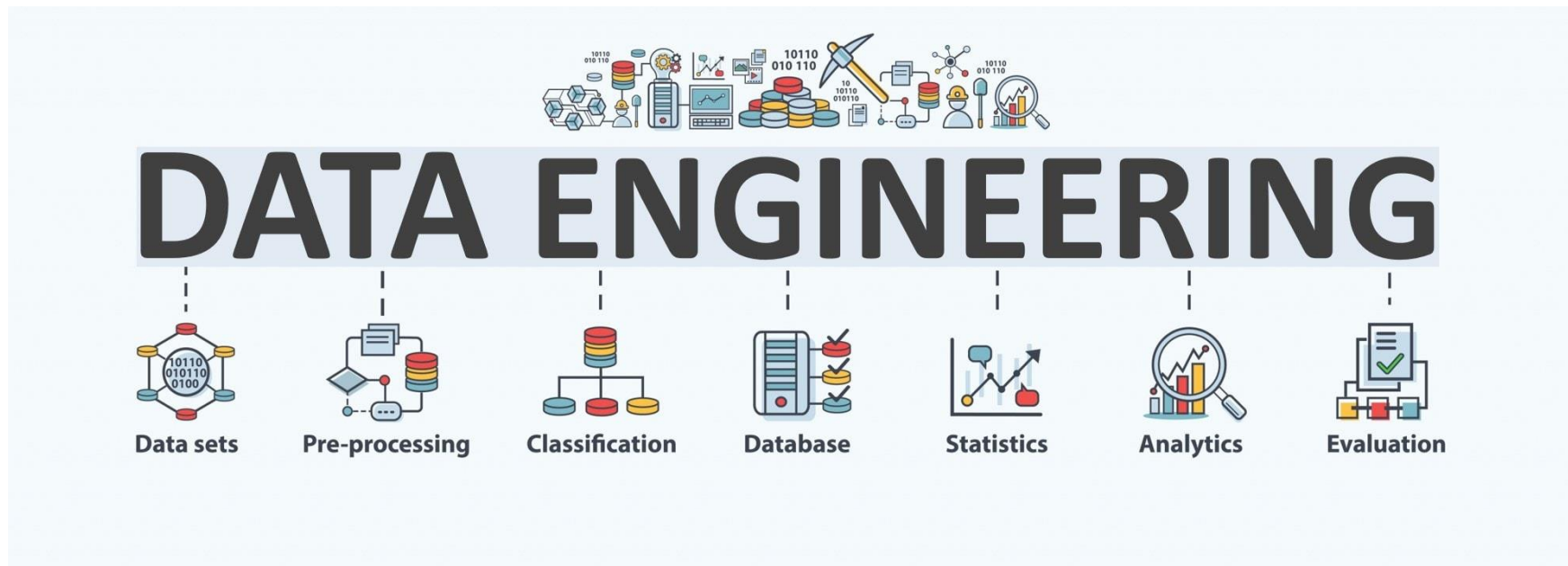


INTRODUCTION TO DATA ENGINEERING



DEFINITION OF DATA ENGINEERING

- Data engineering is the process of designing and building systems that let people collect and analyze raw data from multiple sources and formats. These systems empower people to find practical applications of the data, which businesses can use to thrive.



ROLES AND RESPONSIBILITIES OF A DATA ENGINEER

- Organizing raw data, performing data cleansing, shaping, and ETL processes.
- Collecting raw data from various sources.
- Collecting structured and unstructured data.
- Finding ways to improve data quality and establish data reliability.
- Building data systems and data pipelines.
- Preparing data for model creation.
- Creating algorithms and prototypes.
- Assessing business requirements and objectives.
- Analyzing complex data and reporting results.
- Developing tools and programs for data analysis.

DATA ENGINEERING TOOLS AND TECHNOLOGIES

- Database
- Data Warehouse
- Data Lake
- ETL Tools



DATABASE

Database คือที่สำหรับเก็บข้อมูลที่มีโครงสร้างชัดเจน (**Structured Data**) จากแหล่งข้อมูลหนึ่งไว้ในที่เดียวกัน สามารถเก็บ ค้นหา และบอกเราได้ว่าอะไรอยู่ในนั้น

นอกจากนี้ยังเป็นที่เก็บข้อมูลที่สร้างได้ง่ายที่สุด ใช้ภาษา **SQL** ในการเรียกข้อมูล มักจะถูกใช้ในการทำรายงานทางด้านการเงินและอื่นๆ วิเคราะห์ข้อมูลขนาดเล็ก ทำให้กระบวนการทางธุรกิจกลายเป็นระบบอัตโนมัติ และตรวจสอบการเพิ่มข้อมูล



DATA WAREHOUSE

Data Warehouse คือที่เก็บขนาดใหญ่สำหรับข้อมูลที่มีโครงสร้างชัดเจนจากหลายแหล่ง มารวมกันไว้ โดยที่บริษัทขนาดกลางไปจนถึงขนาดใหญ่มักจะมีไว้ใช้ในการแชร์ข้อมูลระหว่างทีม หรือระหว่างแผนก นักวิเคราะห์ธุรกิจ (**Business Analyst**) สามารถดึง **insight** ออกมา ทำเป็นรายงานไว้สำหรับการตัดสินใจด้านธุรกิจของฝั่งบริหาร



DATA LAKE



Data Lake คือที่เก็บขนาดใหญ่ที่สามารถเก็บข้อมูลได้ทุกรูปแบบจากหลายแหล่งโดยที่ไม่ต้องมีการแปลงข้อมูลก่อน พูดให้เข้าใจง่าย ๆ ก็คือสามารถเก็บข้อมูลดิบได้ ตั้งแต่ข้อมูลที่มีโครงสร้างชัดเจน (**Structured Data**) ข้อมูลกึ่งโครงสร้าง (**Semi-Structured Data**) และข้อมูลที่ไม่มีโครงสร้างแน่นอน (**Unstructured Data**)

ซึ่งสามารถเปลี่ยนรูปแบบโครงสร้างของข้อมูลให้ นักวิทยาศาสตร์ข้อมูล (**Data Scientist**) หรือ นักวิเคราะห์ข้อมูล (**Data Analyst**) ใช้ในการสร้างโมเดล วิเคราะห์ข้อมูลแบบ **real time** ได้ แต่ก็ไม่สามารถดึงข้อมูลมาใช้สร้างรายงาน หรือข้อสรุปในการตัดสินใจทางธุรกิจได้ง่าย ๆ เหมือนใน **Data Warehouse** เพราะจำเป็นต้องทำให้ข้อมูลอยู่ในรูปแบบที่ง่ายต่อการใช้งานก่อน

ETL TOOL

- Apache Airflow
- IBM InfoSphere DataStage
- Oracle Data Integrator
- Microsoft SQL Server Integration Services (SSIS)

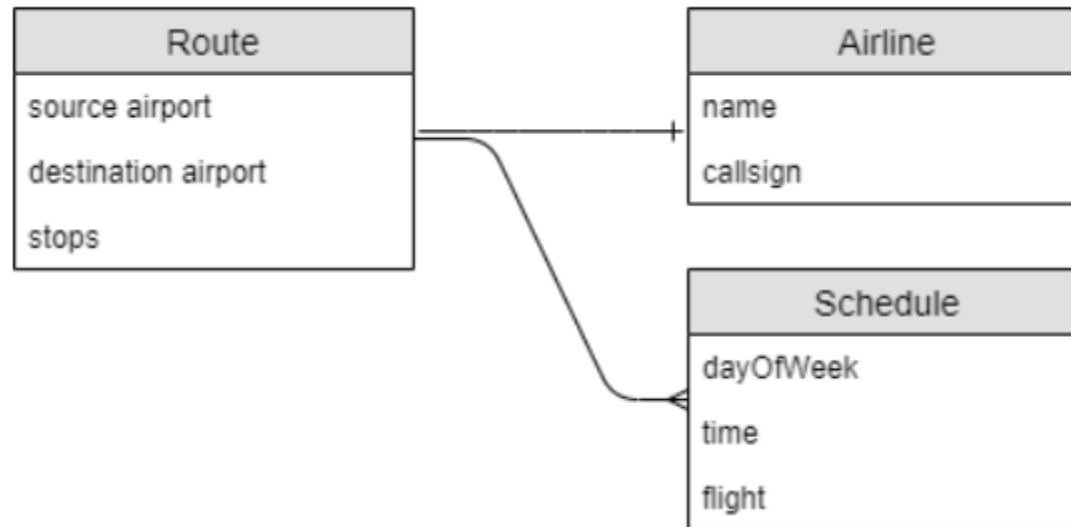




DATA MODELING AND DATABASE DESIGN

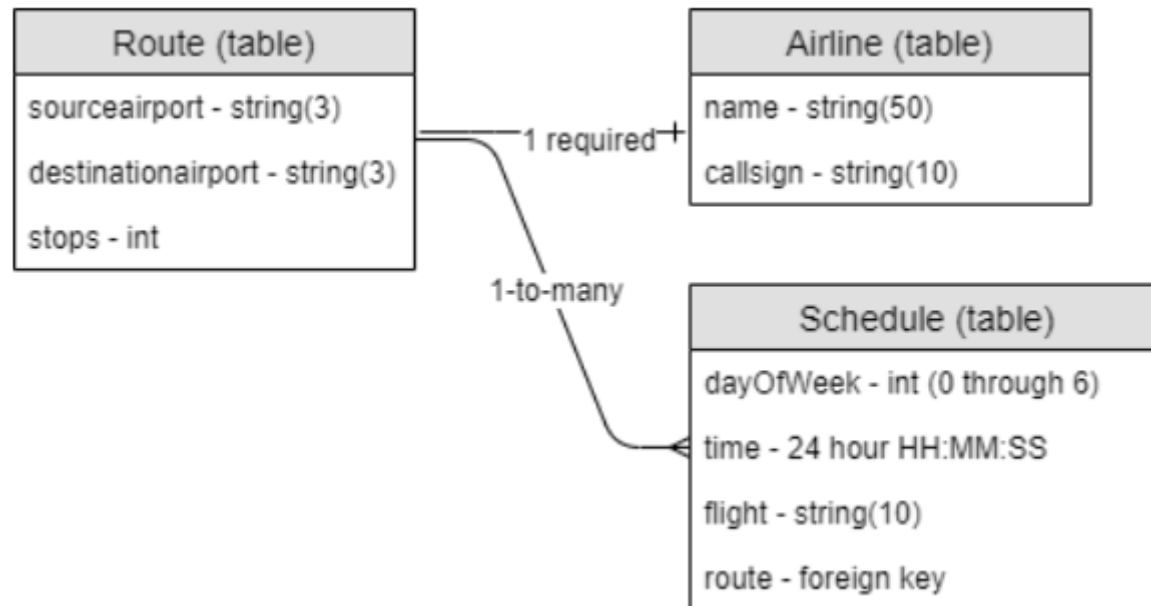
CONCEPTUAL DATA MODELS

- Conceptual Data Model is a visualization of data without considering technical details or the structure of a database system, but focuses on the relationships and linkages of data within an organization.



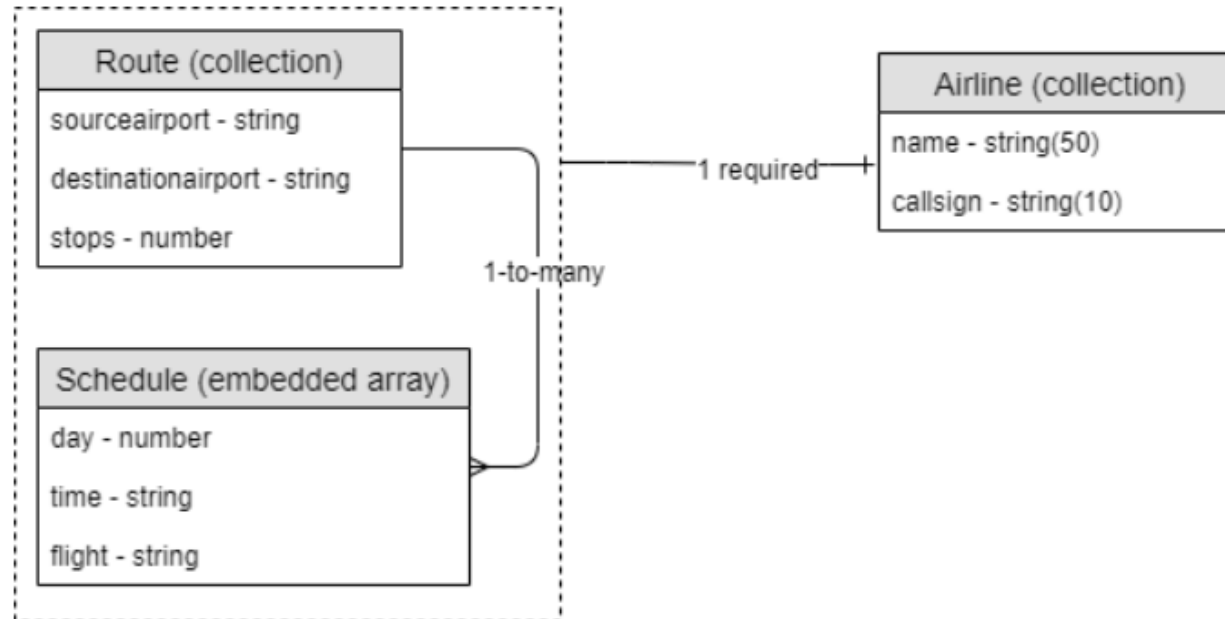
LOGICAL DATA MODELS

- Logical Data Model is a visualization of data that considers technical details and the structure of a database system, which makes it easier to understand and interact with the database.



PHYSICAL DATA MODELS

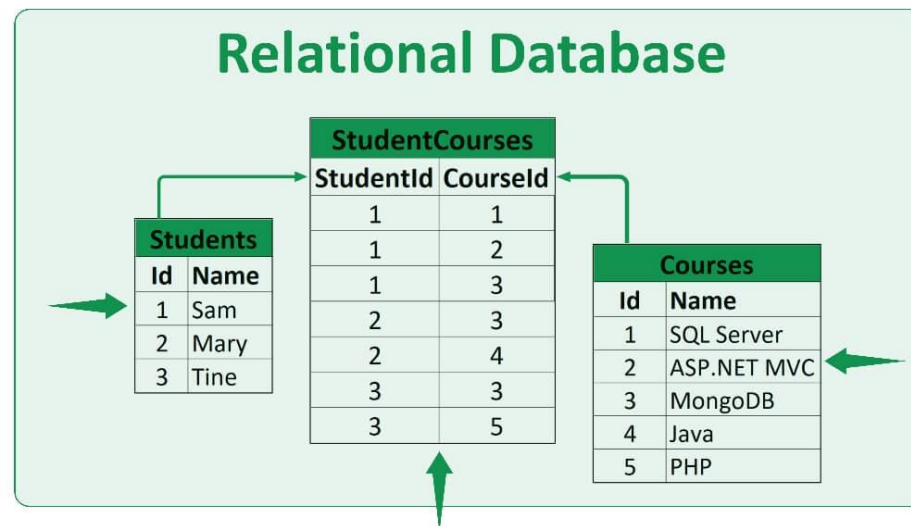
- Physical Data Model is a visualization of data that focuses on technical details and the structure of a database system. It is a process of refining the Logical Data Model to make it applicable for use in a real database system within an organization.



RELATIONAL DATABASES

RDBMS (Relational Database Management System)

เป็นระบบจัดการฐานข้อมูลที่มีความเสถียรมากเหมาะสำหรับการเก็บข้อมูลที่มีจุดประสงค์และแยกประเภทชัดเจน ใช้ภาษา SQL ในการ Query และ Maintain Database มีการเก็บข้อมูลในรูปแบบ Tables (ตาราง) มีองค์ประกอบเป็น Rows และ Columns (มองภาพคล้าย ๆ ตารางของ Microsoft Excel)



RELATIONAL DATABASES

องค์ประกอบของ **Tables** จะประกอบไปด้วย

1. **Row** เรียกอีกชื่อว่า **Tuple** คือ ข้อมูล
2. **Column** เรียกอีกชื่อว่า **Attribute** คือ การระบุชนิดของข้อมูลนั้น ๆ เช่น ที่อยู่, วัน เดือน ปีเกิด
3. **Table** เรียกอีกชื่อว่า **Relation** คือ ชุดของข้อมูลที่แบ่งชนิดเรียบร้อยแล้ว
(Record, Attribute, Rows & Columns)
4. **View** เรียกอีกชื่อว่า **Query** คือ การรายงานข้อมูลจาก **RDBMS** โดยจะเรียกดูจาก **Record** จาก **Row** ใดก็ได้

CONSTRAINTS

Constraints — แปลเป็นไทยก็คือ ข้อบังคับ ซึ่งข้อบังคับนี้จะต้องเกี่ยวข้องกับ **Data Integrity** (ความถูกต้องสมบูรณ์ของข้อมูล) ซึ่งมีอยู่ 2 ส่วน ได้แก่

1.Entity Integrity ต้องสามารถระบุถึงแถวข้อมูลภายใน **Table** ได้

2.Referential Integrity ข้อมูลที่ใส่เข้าไปจะต้องมีความสัมพันธ์กับอีกตารางหนึ่ง (ต้องนำค่ามาจากอีกตารางที่อ้างอิง)

CONSTRAINTS

Key มี 2 รูปแบบ

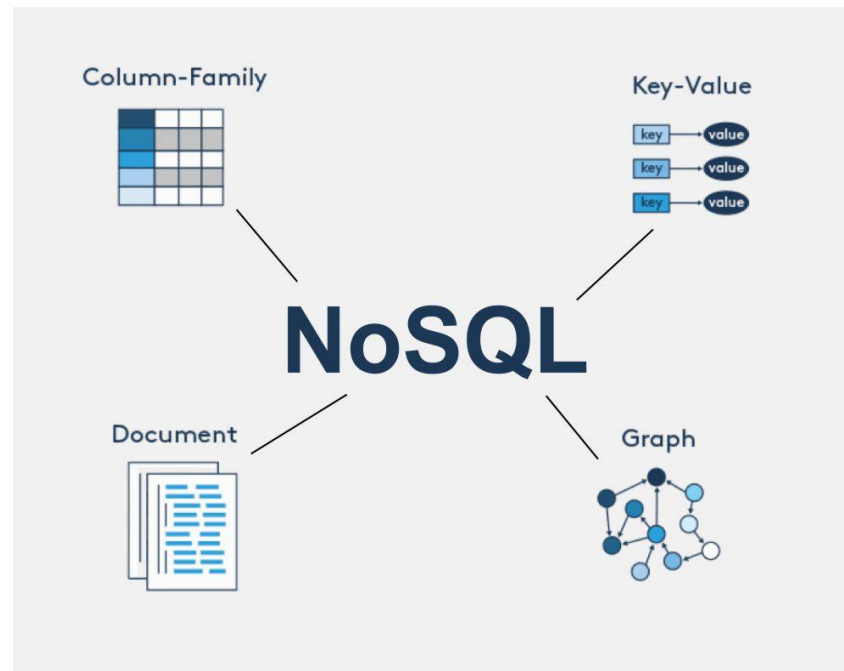
- **Primary Key** หมายถึง จะไม่ให้ใน **Column** มีข้อมูลที่ซ้ำกันและข้อมูลที่ว่างอยู่ (NULL)
- **Foreign Key** หมายถึง ต้องมีการ **Reference** ข้อมูลจาก **Table** ที่มี **Primary Key**

Other

- **NOT NULL** หมายถึง ใน **Record** ต้องมีข้อมูล ถ้าเราไม่ได้กรอก **Record** จะบันทึกเป็น **NULL** โดยอัตโนมัติ
- **UNIQUE** หมายถึง ใน **Table** จะต้องไม่มีข้อมูลที่ซ้ำกัน
- **DEFAULT** หมายถึง จะระบุข้อมูลลงใน **Record** ให้ ถ้าไม่ได้กรอกข้อมูลลงไป
- **CHECK** หมายถึง จะตรวจสอบข้อมูลให้ว่าตรงตามเงื่อนไขหรือไม่

NOSQL DATABASES

Non-SQL ย่อมาจาก Non-relational database เป็น Database อื่น ๆ ที่ไม่ได้เป็นแบบ Relational หรือมีความสัมพันธ์กันชัดเจนแบบ Pattern เหมาะสำหรับการใช้งานจำพวก Big Data และ Real-time Web Application



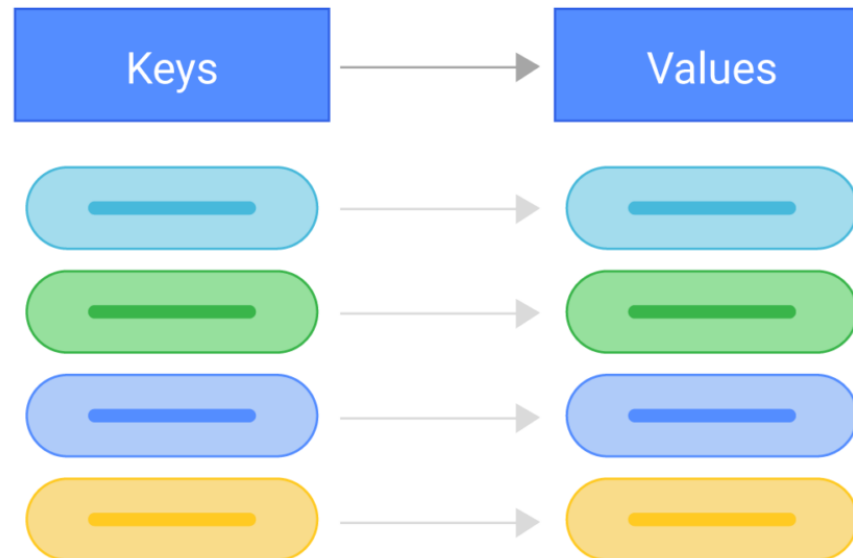
NOSQL DATABASES

- Document ข้อมูลและ Metadata จะเก็บเป็นลำดับชั้นในรูปแบบ Semi-structure data เช่น JSON หรือ XML ใน Database ตัวอย่าง Database Software ที่ใช้งานลักษณะนี้ ได้แก่ Cosmos DB, IBM Domino, MongoDB, Couchbase, ArangoDB



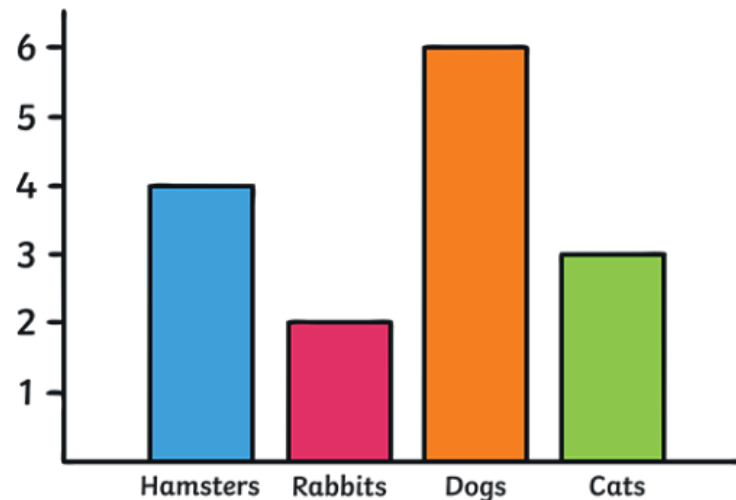
NOSQL DATABASES

- **Key-Value** เป็นการเก็บ **Record** ที่ไม่มีอะไรซับซ้อน มีแค่ **Key** และ **Value** ทำให้สามารถเข้าถึงข้อมูลได้รวดเร็ว โดยการเข้าถึงข้อมูลก็ให้ใช้ **Key** ก็จะได้ **Value** ที่ต้องการ ตัวอย่าง **Database Software** ที่ใช้งานลักษณะนี้ ได้แก่ Redis, Memcached, Apache Ignite, Couchbase, Dynamo



NOSQL DATABASES

- **Graph** ข้อมูลจะเก็บอยู่ในรูปแบบกราฟแผนภูมิ มี **Node** และ **Edge** ที่เชื่อมต่อกัน ทำให้ไม่ต้องนำข้อมูลมา **JOINS** กันเหมือนของ **RDBMS** ตัวอย่าง **Database Software** ที่ใช้งานลักษณะนี้ ได้แก่ **ArangoDB, InfiniteGraph, Apache Giraph, MarkLogic, Neo4J, OrientDB, Virtuoso**



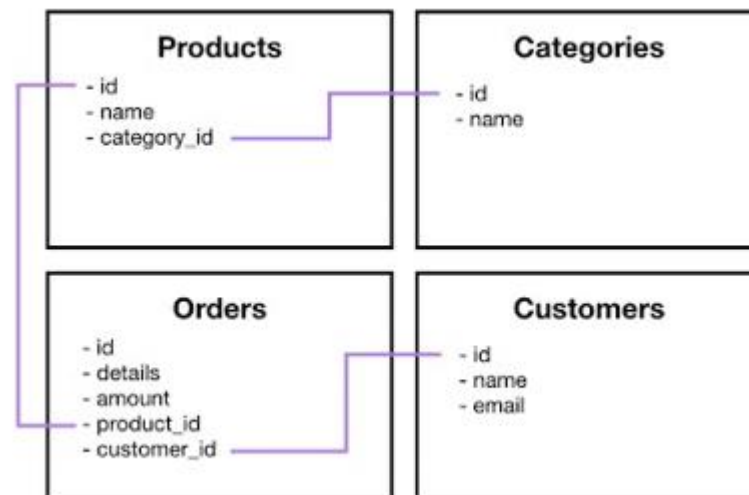
NOSQL DATABASES

- **Wide-Column** รูปแบบของ Wide-Column จะบันทึกข้อมูลในรูปแบบ Tables (Rows และ Columns) แต่จะต่างจาก RDBMS ตรงที่ แต่ละ Rows จะไม่ Fix Column (ถ้าเป็น RDBMS จะ Fix มาเป็น Pattern เดียวกัน) ตัวอย่าง Database Software ที่ใช้งานลักษณะนี้ ได้แก่ Amazon DynamoDB, Cassandra, Azure Tables, Accumulo, HBase

Row A	Column 1	Column 2	Column 3
	Value	Value	Value
Row B	Column 1	Column 2	Column 3
	Value	Value	Value

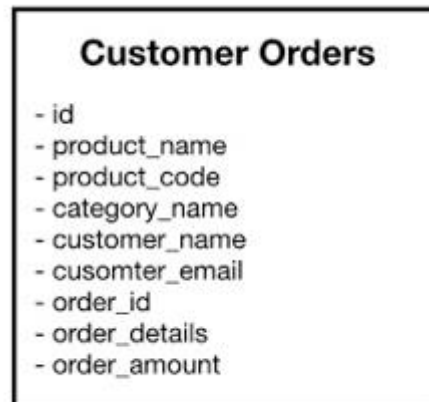
DATA NORMALIZATION

- Normalization involves organizing data into separate tables, each with a single, unique purpose. This helps reduce data redundancy and inconsistencies, and makes it easier to maintain and update the database. The process of normalization typically involves breaking up larger tables into smaller, more specialized ones and establishing relationships between them using foreign keys.



DATA DENORMALIZATION

- Denormalization, on the other hand, involves combining data from multiple tables into a single table to improve query performance. This can be useful in situations where a query requires data from multiple tables, as joining tables can be a time-consuming process. However, denormalization can also lead to data redundancy and make it more difficult to maintain the database.





OVERVIEW OF ETL



ETL (EXTRACT, TRANSFORM, LOAD)

The ETL Process Explained



Extract

Retrieves and verifies data
from various sources



Transform

Processes and organizes
extracted data so it is usable



Load

Moves transformed data
to a data repository

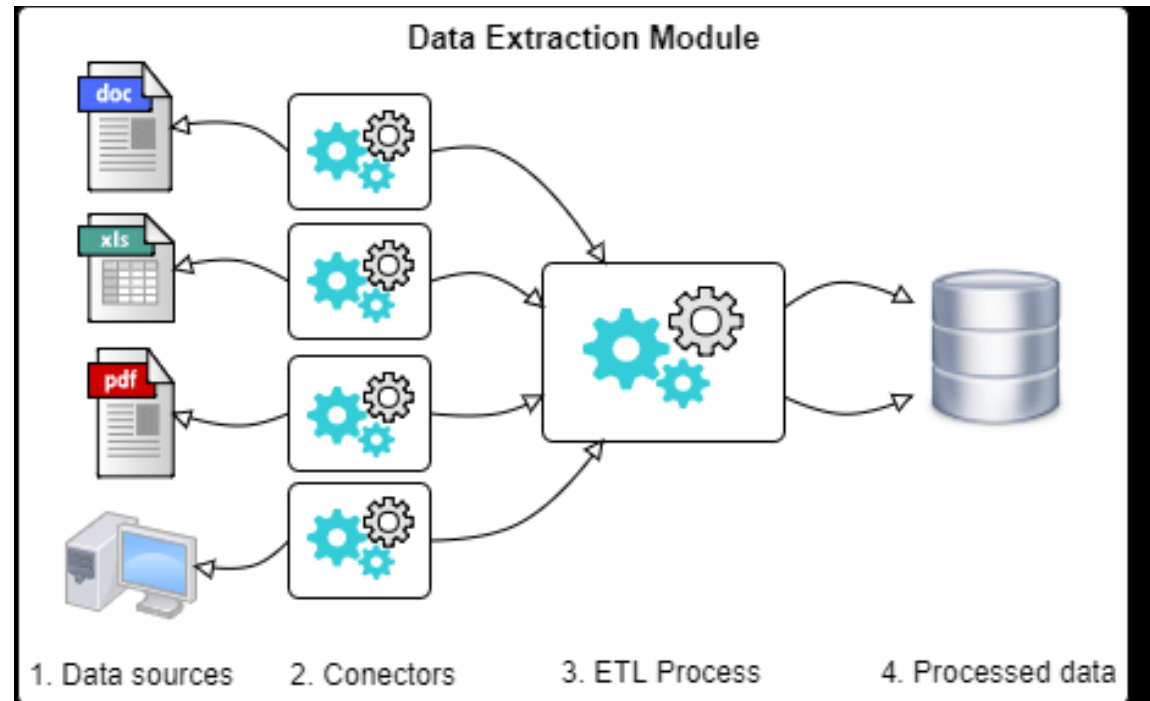
DATA EXTRACTION



- Data extraction is the process of retrieving data from various sources, which can be structured or unstructured, and transforming it into a format that can be used for analysis or storage. The data extraction process typically involves identifying relevant sources of data, selecting and extracting the necessary data elements, and preparing the data for further processing. This can involve cleaning and transforming the data to ensure that it is accurate and consistent. Data extraction is an essential step in data management and is used in a variety of applications, including data warehousing, business intelligence, and data analytics.

EXTRACTING DATA FROM VARIOUS SOURCES

- flat file
- excel
- csv
- database



DATA TRANSFORMATION



- Data transformation refers to the process of converting data from one format, structure, or type to another in order to make it more suitable for analysis, processing, or storage. It involves applying a series of operations or functions to the raw data, such as cleaning, filtering, merging, and aggregating, in order to prepare it for further processing. The goal of data transformation is to create a more structured and consistent dataset that is easier to work with and provides more meaningful insights. Some common tools used for data transformation include scripting languages, database management systems, and data integration platforms.

DATA LOADING



- Data Loading is the process of inserting, updating, or deleting data in a target database, data warehouse, or data lake. This involves moving data from the source system to the destination system, and it is typically the final step in an ETL (Extract, Transform, Load) process. The data is often loaded into a staging area first, where it can be processed, validated, and transformed before being loaded into the final target system. Data loading can be performed in various ways, such as batch processing, real-time processing, or incremental processing. The goal of data loading is to ensure that the data is accurate, consistent, and accessible for analysis and reporting purposes.



SQL RECAP



SELECT

```
select e.employee_id,  
       e.first_name || ' ' || e.last_name Full_name,  
       lower(e.email) || '@a-host.co.th' email,  
       j.job_title, j.Min_salary, j.max_salary  
from employees e  
join jobs j on e.job_id = j.job_id  
order by 1;
```

SELECT

Query Result x						
SQL All Rows Fetched: 107 in 0.048 seconds						
	EMPLOYEE_ID	FULL_NAME	EMAIL	JOB_TITLE	MIN_SALARY	MAX_SALARY
1	100	Steven King	sking@a-host.co.th	President	20080	40000
2	101	Neena Kochhar	nkochhar@a-host.co.th	Administration Vice President	15000	30000
3	102	Lex De Haan	ldehaan@a-host.co.th	Administration Vice President	15000	30000
4	103	Alexander Hunold	ahunold@a-host.co.th	Programmer	4000	10000
5	104	Bruce Ernst	bernst@a-host.co.th	Programmer	4000	10000
6	105	David Austin	daustin@a-host.co.th	Programmer	4000	10000
7	106	Valli Pataballa	vpatabal@a-host.co.th	Programmer	4000	10000
8	107	Diana Lorentz	dlorentz@a-host.co.th	Programmer	4000	10000
9	108	Nancy Greenberg	ngreenbe@a-host.co.th	Finance Manager	8200	16000
10	109	Daniel Faviat	dfaviat@a-host.co.th	Accountant	4200	9000
11	110	John Chen	jchen@a-host.co.th	Accountant	4200	9000
12	111	Ismael Sciarra	isciarra@a-host.co.th	Accountant	4200	9000
13	112	Jose Manuel Urman	jmurman@a-host.co.th	Accountant	4200	9000
14	113	Luis Popp	lpopp@a-host.co.th	Accountant	4200	9000
15	114	Den Raphaely	drapheal@a-host.co.th	Purchasing Manager	8000	15000
16	115	Alexander Khoo	akhoo@a-host.co.th	Purchasing Clerk	2500	5500
17	116	Shelli Baida	sbaida@a-host.co.th	Purchasing Clerk	2500	5500
18	117	Sigal Tobias	stobias@a-host.co.th	Purchasing Clerk	2500	5500

CREATE TABLE/VIEW

```
Create table dummy_employees as  
select * from employees  
where employee_id = 100;
```

CREATE TABLE/VIEW

create table dummy_employees2

```
(  
  ▪ EMPLOYEE_CD2                NUMBER(6),  
  ▪ EMPLOYEE_NAME2             NVARCHAR2(200),  
  ▪ EMPLOYEE_EMAIL2            NVARCHAR2(50),  
  ▪ EMPLOYEE_PHONE_NUMBER2     NVARCHAR2(200),  
  ▪ HIRE_DATE2                 DATE,  
  ▪ LAST_DATE2                 NVARCHAR2(50),  
  ▪ JOB_TITLE2                 NVARCHAR2(35),  
  ▪ EMPLOYEE_COMMISSION2       NUMBER(2,2),  
  ▪ MANAGER_CD2                NUMBER(6),  
  ▪ MANAGER_NAME2              NVARCHAR2(200),  
  ▪ ETL_DATE2                  TIMESTAMP(6),  
  ▪ ETL_LAST_UPDATE2           TIMESTAMP(6)  
);
```

CREATE TABLE/VIEW

```
Script Output x Query Result x
Task completed in 0.034 seconds

Table DUMMY_EMPLOYEES created.

Table DUMMY_EMPLOYEES2 created.
```

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	100	Steven	King	SKING	515.123.4567	17-JUN-03	AD_PRES	24000	(null)	(null)	90

EMPLOY...	EMPLOY...	EMPLOY...	EMPLOY...	HIRE_D...	LAST_D...	JOB_TITLE	EMPLOY...	MANAGE...	MANAGE...	ETL_DATE	ETL_LA...
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	----------	-----------

CREATE TABLE/VIEW

```
create or replace view v_dummy_employees as  
select * from employees;
```

CREATE TABLE/VIEW

```
create or replace view v_dummy_employees2 as
select e.employee_id,
       e.first_name || ' ' || e.last_name Full_name,
       lower(e.email) || '@a-host.co.th' email,
       j.job_title, j.Min_salary, j.max_salary
from employees e
join jobs j on e.job_id = j.job_id
order by 1;
```


CREATE TABLE/VIEW

```
Script Output x Query Result x
Task completed in 0.039 seconds

View V_DUMMY_EMPLOYEES created.

View V_DUMMY_EMPLOYEES2 created.
```

Script Output x Query Result x

 SQL | Fetched 50 rows in 0.004 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	100	Steven	King	SKING	515.123.4567	17-JUN-03	AD_PRES	24000	(null)	(null)	90
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-05	AD_VP	17000	(null)	100	90
3	102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-01	AD_VP	17000	(null)	100	90
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-06	IT_PROG	9000	(null)	102	60
5	104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-07	IT_PROG	6000	(null)	103	60
6	105	David	Austin	DAUSTIN	590.423.4569	25-JUN-05	IT_PROG	4800	(null)	103	60
7	106	Valli	Pataballa	VPATABAL	590.423.4560	05-FEB-06	IT_PROG	4800	(null)	103	60
8	107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-07	IT_PROG	4200	(null)	103	60
9	108	Nancy	Greenberg	NGREENBE	515.124.4569	17-AUG-02	FI_MGR	12008	(null)	101	100
10	109	Daniel	Faviet	DFAVIET	515.124.4169	16-AUG-02	FI_ACCOUNT	9000	(null)	108	100
11	110	John	Chen	JCHEN	515.124.4269	28-SEP-05	FI_ACCOUNT	8200	(null)	108	100
12	111	Ismael	Sciarra	ISCIARRA	515.124.4369	30-SEP-05	FI_ACCOUNT	7700	(null)	108	100
13	112	Jose Manuel	Urman	JMURMAN	515.124.4469	07-MAR-06	FI_ACCOUNT	7800	(null)	108	100
14	113	Luis	Popp	LPOPP	515.124.4567	07-DEC-07	FI_ACCOUNT	6900	(null)	108	100
15	114	Den	Raphaely	DRAPHEAL	515.127.4561	07-DEC-02	FU_MAN	11000	(null)	100	30
16	115	Alexander	Khoo	AKHOO	515.127.4562	18-MAY-03	FU_CLERK	3100	(null)	114	30
17	116	Shelli	Baida	SBAIDA	515.127.4563	24-DEC-05	FU_CLERK	2900	(null)	114	30
18	117	Sigal	Tobias	STOBIAS	515.127.4564	24-JUL-05	FU_CLERK	2800	(null)	114	30

CREATE TABLE/VIEW

Script Output x Query Result x

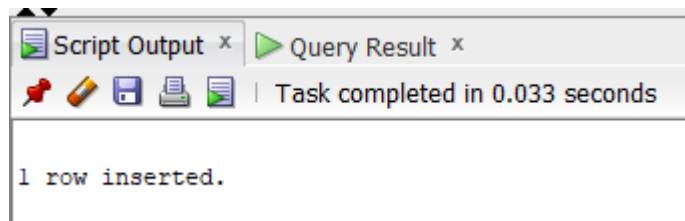
SQL | Fetched 50 rows in 0.004 seconds

	EMPLOYEE_ID	FULL_NAME	EMAIL	JOB_TITLE	MIN_SALARY	MAX_SALARY
1	100	Steven King	sking@a-host.co.th	President	20080	40000
2	101	Neena Kochhar	nkochhar@a-host.co.th	Administration Vice President	15000	30000
3	102	Lex De Haan	ldehaan@a-host.co.th	Administration Vice President	15000	30000
4	103	Alexander Hunold	ahunold@a-host.co.th	Programmer	4000	10000
5	104	Bruce Ernst	bernst@a-host.co.th	Programmer	4000	10000
6	105	David Austin	daustin@a-host.co.th	Programmer	4000	10000
7	106	Valli Pataballa	vpatabal@a-host.co.th	Programmer	4000	10000
8	107	Diana Lorentz	dlorentz@a-host.co.th	Programmer	4000	10000
9	108	Nancy Greenberg	ngreenbe@a-host.co.th	Finance Manager	8200	16000
10	109	Daniel Faviyet	dfaviyet@a-host.co.th	Accountant	4200	9000
11	110	John Chen	jchen@a-host.co.th	Accountant	4200	9000
12	111	Ismael Sciarra	isciarra@a-host.co.th	Accountant	4200	9000
13	112	Jose Manuel Urman	jmurman@a-host.co.th	Accountant	4200	9000
14	113	Luis Popp	lpopp@a-host.co.th	Accountant	4200	9000
15	114	Den Raphaely	drapheal@a-host.co.th	Purchasing Manager	8000	15000
16	115	Alexander Khoo	akhoo@a-host.co.th	Purchasing Clerk	2500	5500
17	116	Shelli Baida	sbaida@a-host.co.th	Purchasing Clerk	2500	5500
18	117	Sigal Tobias	stobias@a-host.co.th	Purchasing Clerk	2500	5500

INSERT

```
insert into dummy_employees2  
VALUES (999,'JOEY SIRAPRUET',  
        'sirapruet@a-host.co.th','000.000.0000',  
        '01-MAR-2022','16-JUN-2023',  
        'Hero',null,null,null,null,null);
```

INSERT



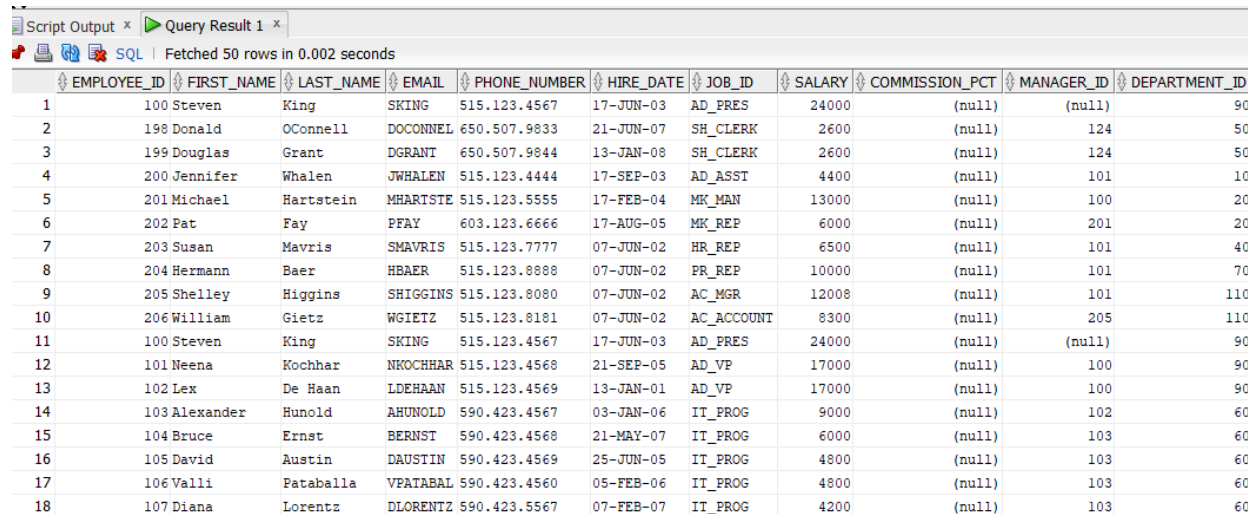
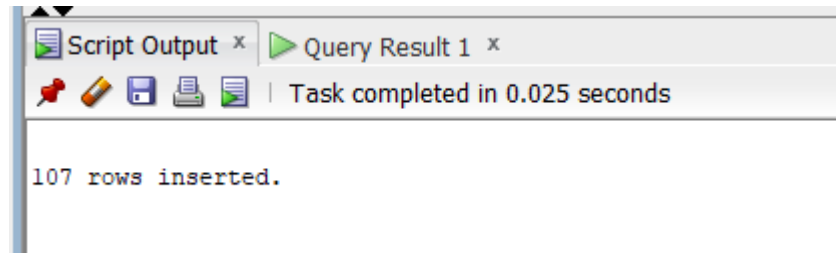
A screenshot of a SQL query result window. The window has a title bar with 'Script Output' and 'Query Result' tabs. Below the title bar, there are icons for a red pin, a yellow pencil, a blue floppy disk, a green printer, and a green document. To the right of these icons, it says 'All Rows Fetched: 1 in 0.001 seconds'. The main area of the window displays a table with 12 columns and 1 row. The columns are: EMPLOYEE_CD, EMPLOYEE_NAME, EMPLOYEE_EMAIL, EMPLOYEE_PHONE_NUMBER, HIRE_DATE, LAST_DATE, JOB_TITLE, EMPLOYEE_COMMISSION, MANAGER_CD, MANAGER_NAME, ETL_DATE, and ETL_LAST_UPDATE. The row contains the following values: 1, 999 JOEY SIRAFRUET, sirapruet@a-host.c..., 000.000.0000, 01-MAR-22, 16-JUN-2023, Hero, (null), (null), (null), (null), and (null).

EMPLOYEE_CD	EMPLOYEE_NAME	EMPLOYEE_EMAIL	EMPLOYEE_PHONE_NUMBER	HIRE_DATE	LAST_DATE	JOB_TITLE	EMPLOYEE_COMMISSION	MANAGER_CD	MANAGER_NAME	ETL_DATE	ETL_LAST_UPDATE
1	999 JOEY SIRAFRUET	sirapruet@a-host.c...	000.000.0000	01-MAR-22	16-JUN-2023	Hero	(null)	(null)	(null)	(null)	(null)

INSERT

```
insert into dummy_employees  
select * from employees;
```

INSERT



Script Output x Query Result 1 x

SQL | Fetched 50 rows in 0.002 seconds

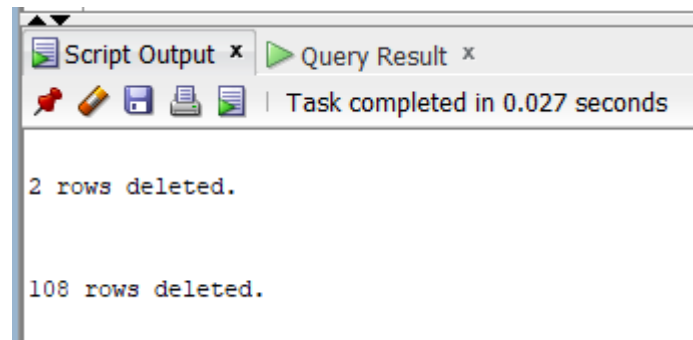
	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	100	Steven	King	SKING	515.123.4567	17-JUN-03	AD_PRES	24000	(null)	(null)	90
2	198	Donald	OConnell	DOCONNEL	650.507.9833	21-JUN-07	SH_CLERK	2600	(null)	124	50
3	199	Douglas	Grant	DGRANT	650.507.9844	13-JAN-08	SH_CLERK	2600	(null)	124	50
4	200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-03	AD_ASST	4400	(null)	101	10
5	201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-04	MK_MAN	13000	(null)	100	20
6	202	Pat	Fay	PFAY	603.123.6666	17-AUG-05	MK_REP	6000	(null)	201	20
7	203	Susan	Mavris	SMAVRIS	515.123.7777	07-JUN-02	HR_REP	6500	(null)	101	40
8	204	Hermann	Baer	HBAER	515.123.8888	07-JUN-02	PR_REP	10000	(null)	101	70
9	205	Shelley	Higgins	SHIGGINS	515.123.8080	07-JUN-02	AC_MGR	12008	(null)	101	110
10	206	William	Gietz	WGIEZT	515.123.8181	07-JUN-02	AC_ACCOUNT	8300	(null)	205	110
11	100	Steven	King	SKING	515.123.4567	17-JUN-03	AD_PRES	24000	(null)	(null)	90
12	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-05	AD_VP	17000	(null)	100	90
13	102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-01	AD_VP	17000	(null)	100	90
14	103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-06	IT_PROG	9000	(null)	102	60
15	104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-07	IT_PROG	6000	(null)	103	60
16	105	David	Austin	DAUSTIN	590.423.4569	25-JUN-05	IT_PROG	4800	(null)	103	60
17	106	Valli	Pataballa	VPATABAL	590.423.4560	05-FEB-06	IT_PROG	4800	(null)	103	60
18	107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-07	IT_PROG	4200	(null)	103	60

DELETE

```
delete from dummy_employees  
where employee_id = 100;
```

```
delete from dummy_employees;
```

DELETE





ETL LAB(SQL) |

WORKSHOP



ETL-Workshop.txt