Intro to SQL

Fernando Rios, PhD.
Research Data Management Specialist, UA Libraries

May 2, 2025

About me

- Academic
  - Geographic information systems & science
  - Computational hydrogeology
  - Computational physics
- IT
  - Software development
  - Database administration
- Research data management
  - Data organization, metadata
  - Data management plans
  - Data policy compliance
- Manage U of A’s research data repository, ReDATA
  - Funder and journal data sharing requirements
  - University data retention policies
  - Data curation for more reusable data
- Teaching

Data Storage – Un- and semi-structured information

- Measurements about stuff, information about a groups of things, time series
  - Physical lab notebooks
  - Images, videos
  - Individual tables in an unstructured spreadsheet
  - Information in proprietary formats

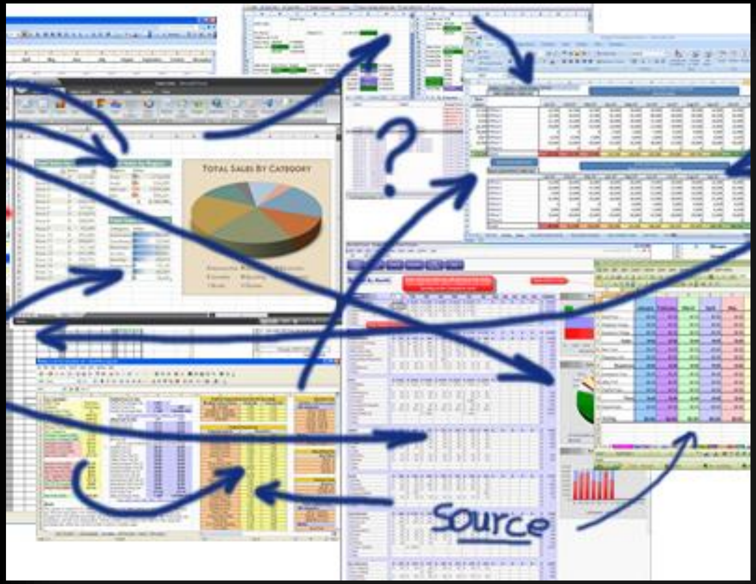
- Unorganized – not easily machine-readable
- Limited analysis capabilities
- Doesn’t scale well

Structured Spreadsheets

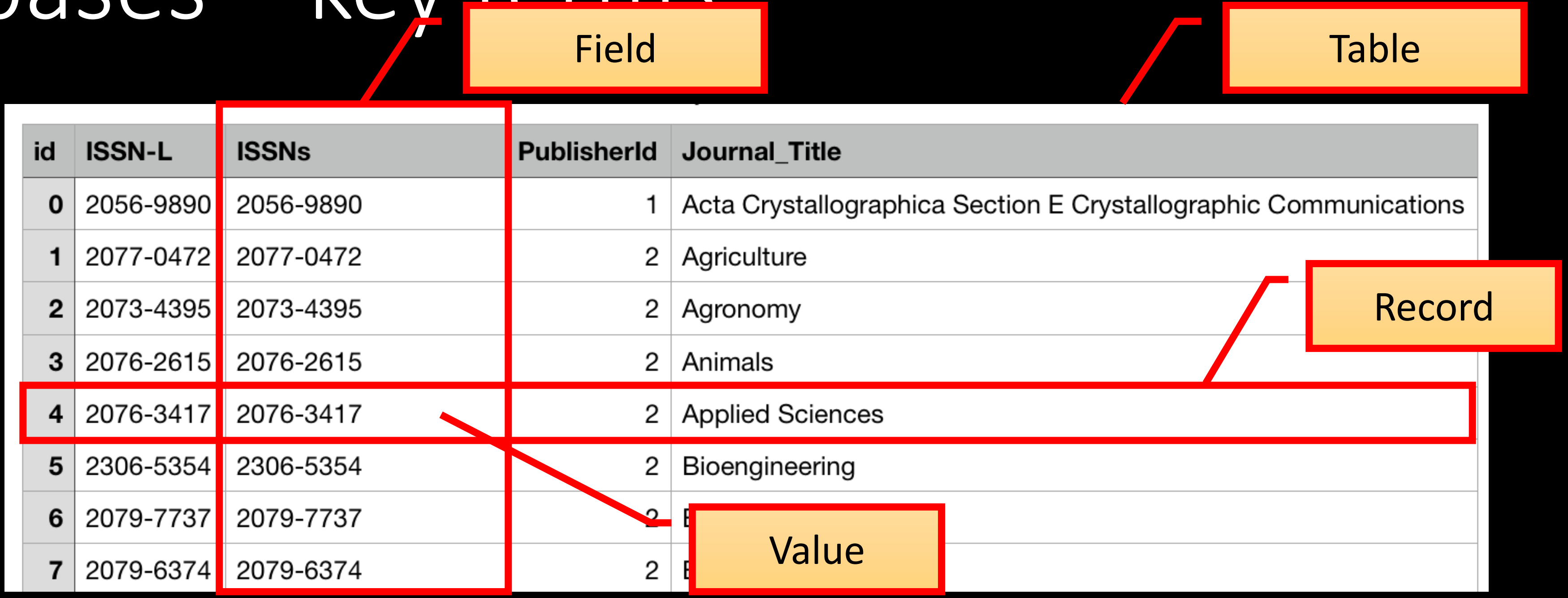
- Standardizes item characteristics, records them in a tabular structure
- Few constraints = Need to follow best-practices https://osf.io/vew32
- Calculations in Excel error-prone as complexity increases
- When to use
  - Simple column/row summaries
  - Doing a quick filter or plot
  - Simple column calculations / one-off pivot tables
  - Exploring data
  - Low consequences

Databases

- Adds more structure, constraints
- Compact “formulas” and repeatable processes = always desirable
- When to use
  - Data can be reasonably put in a table format
  - Analyzing tens of thousands of rows
  - When your formulas start getting too complex
  - Need to ensure data integrity
  - Management capabilities



Databases – key terms



id	ISSN-L	ISSNs	PublisherId	Journal_Title
0	2056-9890	2056-9890	1	Acta Crystallographica Section E Crystallographic Communications
1	2077-0472	2077-0472	2	Agriculture
2	2073-4395	2073-4395	2	Agronomy
3	2076-2615	2076-2615	2	Animals
4	2076-3417	2076-3417	2	Applied Sciences
5	2306-5354	2306-5354	2	Bioengineering
6	2079-7737	2079-7737	2	B
7	2079-6374	2079-6374	2	B

- Values: store a single piece of information
- Fields: single kind and type of information
  - temperature, age, address, etc.
  - integer, text, date, etc.
- Record: set of related fields containing specific values
- Usually have more than one table that are related

SQL, RDBMS?

- Structured Query Language (SQL)
- Relational database systems (RDBMS)
  - Structured relationships between tables + management layer
  - Use SQL to manipulate data in the DB
  - More common in business than science
    - Metadata could be in a DB – even if actual data isn't

SQLite

MySQL™

A

ORACLE®

PostgreSQL

PostGIS

Spatial PostgreSQL

What can we do with RDBMS + SQL?

- Aggregating, summarizing, combining, filtering, adding
- Robust, reproducible
- Ensure better data quality
- Scalable and fast
- Management capabilities
  - Access controls
  - Concurrency
  - Auditing
  - Backups
  - Distributed data

Other kinds of databases

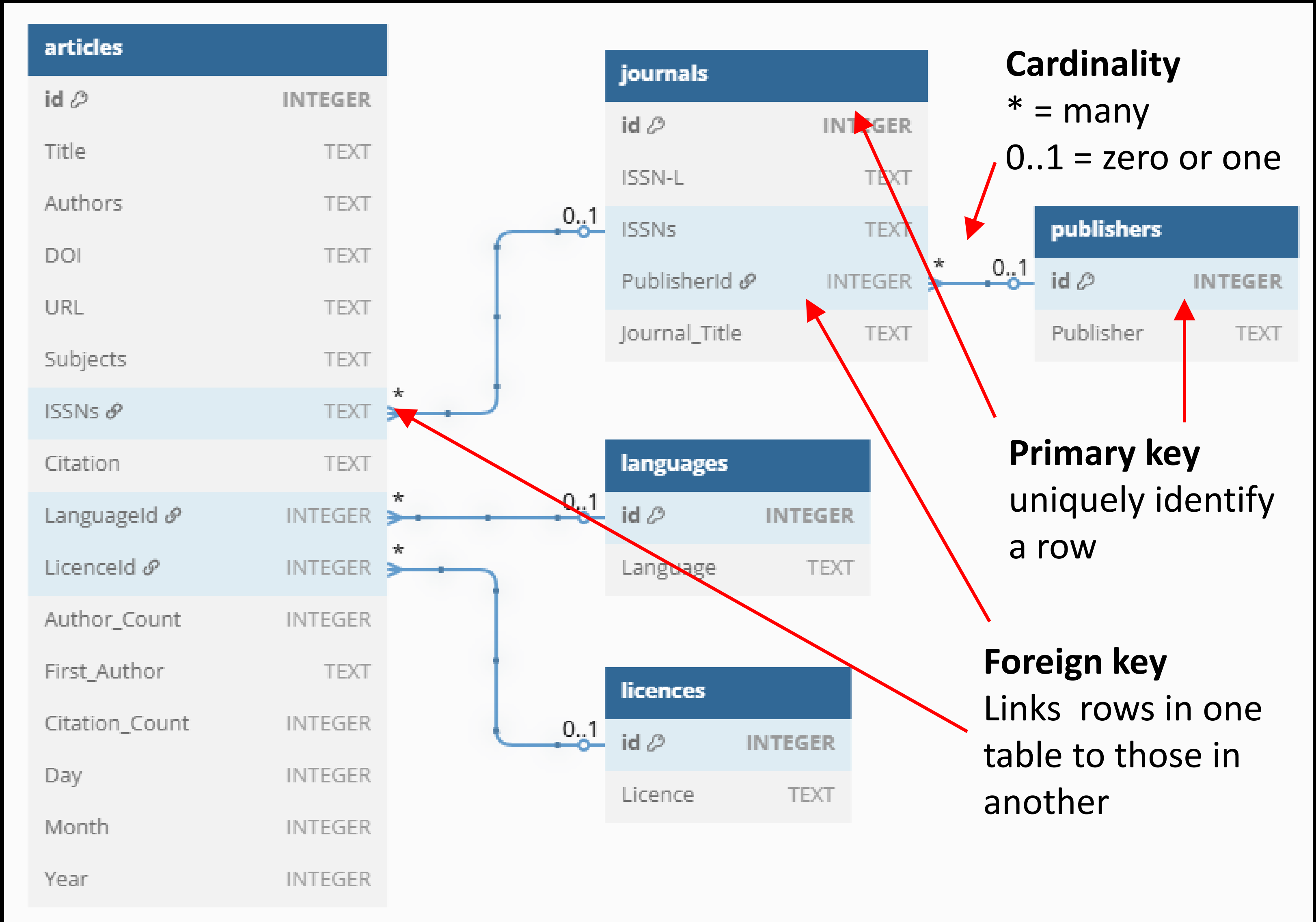
- Key-value
  - Simple, fast
  - Python dictionaries, Redis
- Document-based
  - Less structured, more heterogeneous
  - MongoDB
- Graph databases
  - Entities (nodes), predicates (edge), objects (another node). Complex relationships, knowledge graphs
  - Node4j
- Wide-column
  - Rows can have different columns within the same table
  - Google BigTable

Let’s get started

https://tinyurl.com/s4u42xb3

Relationships – key concepts

Database schema



articles	
id	INTEGER
Title	TEXT
Authors	TEXT
DOI	TEXT
URL	TEXT
Subjects	TEXT
ISSNs	TEXT
Citation	TEXT
LanguageId	INTEGER
LicenceId	INTEGER
Author_Count	INTEGER
First_Author	TEXT
Citation_Count	INTEGER
Day	INTEGER
Month	INTEGER
Year	INTEGER

journals	
id	INTEGER
ISSN-L	TEXT
ISSNs	TEXT
PublisherId	INTEGER
Journal_Title	TEXT

publishers	
id	INTEGER
Publisher	TEXT

languages	
id	INTEGER
Language	TEXT

licences	
id	INTEGER
Licence	TEXT

Cardinality
* = many
0..1 = zero or one

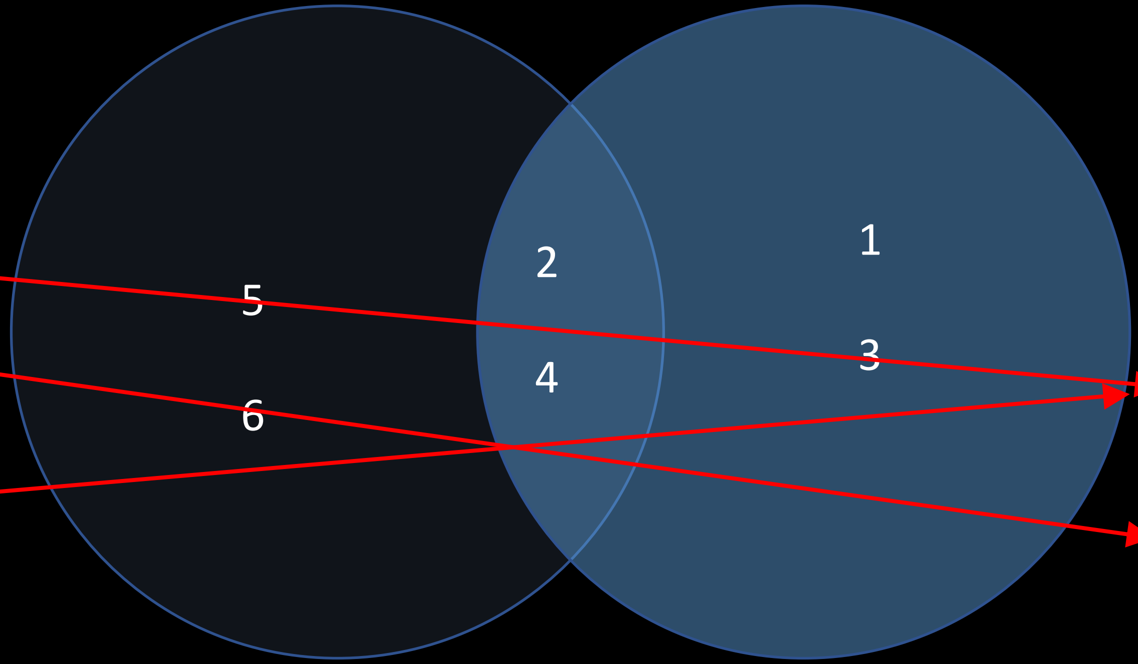
Primary key
uniquely identify a row

Foreign key
Links rows in one table to those in another

Joins

Journals

Journal	PubID
ISPRS	2
JDDT	4
JDMDS	5
JLPEA	2
OPCJ	6



Publishers

Id	Name
1	Consejo Superior de Investigaciones Científicas
2	MDPI AG
3	Soc. of Pharmaceutical Technocrats
4	Int. Union of Crystallography

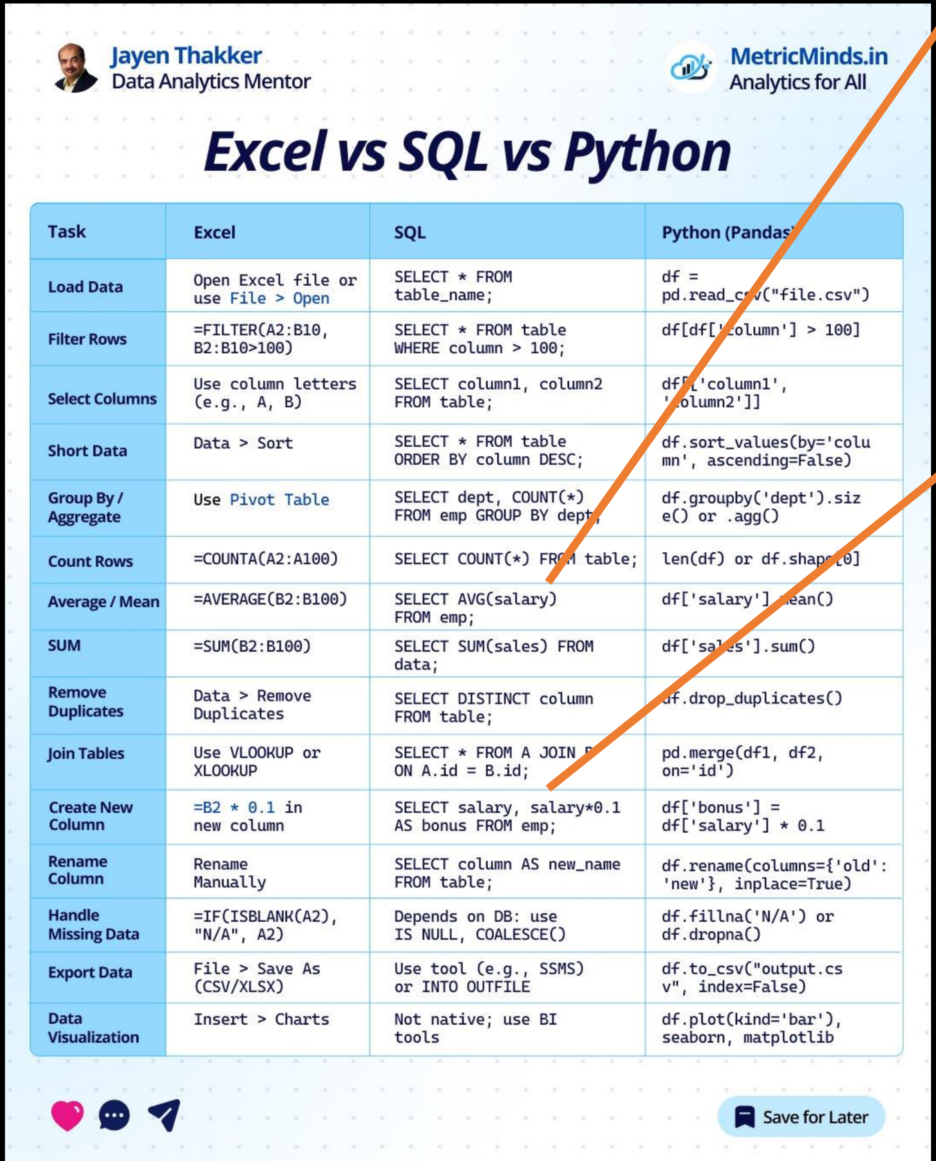
(Inner) Join

Journal	PubID	Id	Name
ISPRS	2	2	MDPI AG
JDDT	4	4	Int. Union of Crystallography
JLPEA	2	2	MDPI AG

Key points of RDBMS

- Data integrity
  - Keeps data separate from analysis
  - Reduces accidents
  - Data types help with quality control
- Data is stored in related tables
  - Reduce redundancy, increase data quality
- Many RDBMS with different strengths
  - SQLite – small compact. Embedded devices
  - MySQL, PostgreSQL – general purpose
  - MS SQL Server, Oracle – Strong management capabilities, enterprise solution

Do I have to use SQL?



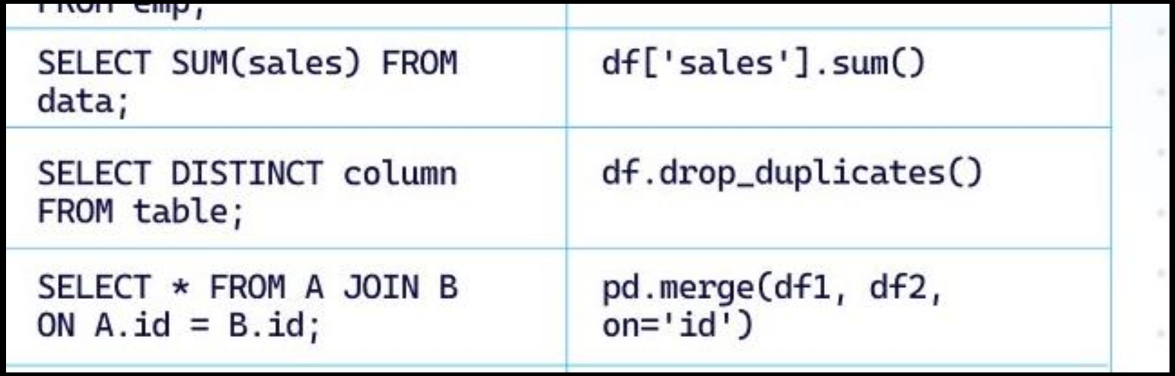
Jayen Thakker
Data Analytics Mentor

MetricMinds.in
Analytics for All

Excel vs SQL vs Python

Task	Excel	SQL	Python (Pandas)
Load Data	Open Excel file or use File > Open	SELECT * FROM table_name;	df = pd.read_csv("file.csv")
Filter Rows	=FILTER(A2:B10, B2:B10>100)	SELECT * FROM table WHERE column > 100;	df[df['column'] > 100]
Select Columns	Use column letters (e.g., A, B)	SELECT column1, column2 FROM table;	df[['column1', 'column2']]
Short Data	Data > Sort	SELECT * FROM table ORDER BY column DESC;	df.sort_values(by='column', ascending=False)
Group By / Aggregate	Use Pivot Table	SELECT dept, COUNT(*) FROM emp GROUP BY dept;	df.groupby('dept').size() or .agg()
Count Rows	=COUNTA(A2:A100)	SELECT COUNT(*) FROM table;	len(df) or df.shape[0]
Average / Mean	=AVERAGE(B2:B100)	SELECT AVG(salary) FROM emp;	df['salary'].mean()
SUM	=SUM(B2:B100)	SELECT SUM(sales) FROM data;	df['sales'].sum()
Remove Duplicates	Data > Remove Duplicates	SELECT DISTINCT column FROM table;	df.drop_duplicates()
Join Tables	Use VLOOKUP or XLOOKUP	SELECT * FROM A JOIN B ON A.id = B.id;	pd.merge(df1, df2, on='id')
Create New Column	=B2 * 0.1 in new column	SELECT salary, salary*0.1 AS bonus FROM emp;	df['bonus'] = df['salary'] * 0.1
Rename Column	Rename Manually	SELECT column AS new_name FROM table;	df.rename(columns={'old': 'new'}, inplace=True)
Handle Missing Data	=IF(ISBLANK(A2), "N/A", A2)	Depends on DB: use IS NULL, COALESCE()	df.fillna('N/A') or df.dropna()
Export Data	File > Save As (CSV/XLSX)	Use tool (e.g., SSMS) or INTO OUTFILE	df.to_csv("output.csv", index=False)
Data Visualization	Insert > Charts	Not native; use BI tools	df.plot(kind='bar'), seaborn, matplotlib

Save for Later



FROM emp;	
SELECT SUM(sales) FROM data;	df['sales'].sum()
SELECT DISTINCT column FROM table;	df.drop_duplicates()
SELECT * FROM A JOIN B ON A.id = B.id;	pd.merge(df1, df2, on='id')