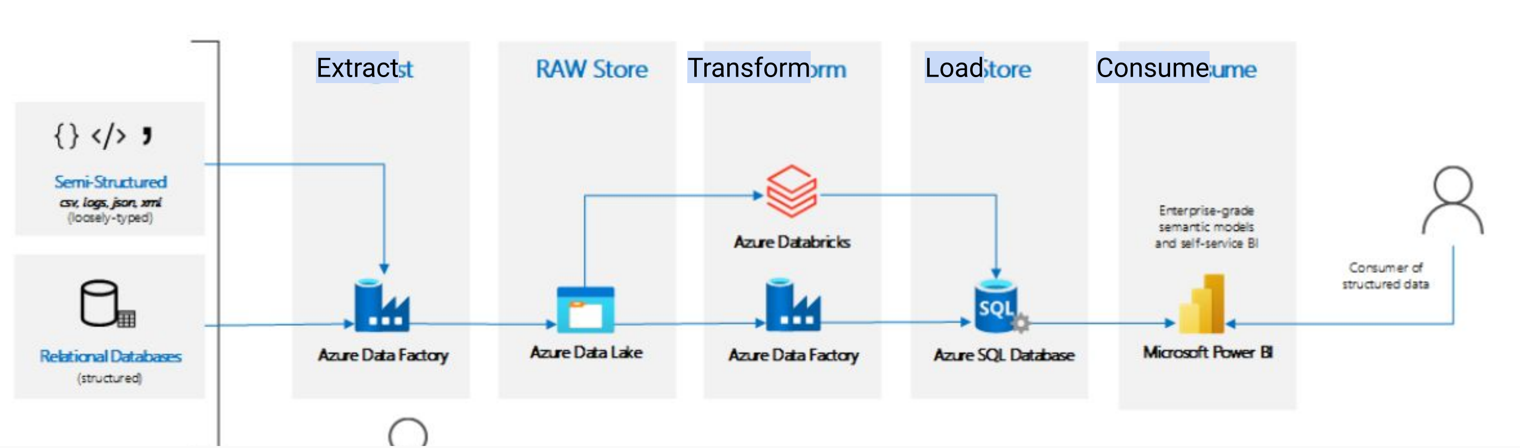
ETL & Analyzing Data

Sabya D.
| Lighthouse Labs Instructor, Data Analytics

Agenda

- Intro to Data Platforms
- ETL vs ELT Paradigms | Why , What & How
- Database Object vs. Schema / Catalogues
- SQL Dialects for ETL
  - Data Definition Languages (DDL)
  - Data Manipulation Languages (DML)
- Demo-1: Use, DDL/DML to load data and analyze
- Introduce: Descriptive Statistics, Common Data types (SQL)
- *Demo-2: Postgres Demo (Optional)*

Intro to Data Platforms |



Concept I

Structured Data : SQL DatabasesMSQL, PostGres SQL
Semi-Structured Data : No-SQL Databases
Un-Structured Data : Movies, Files

Concept II

Datalake vs Database

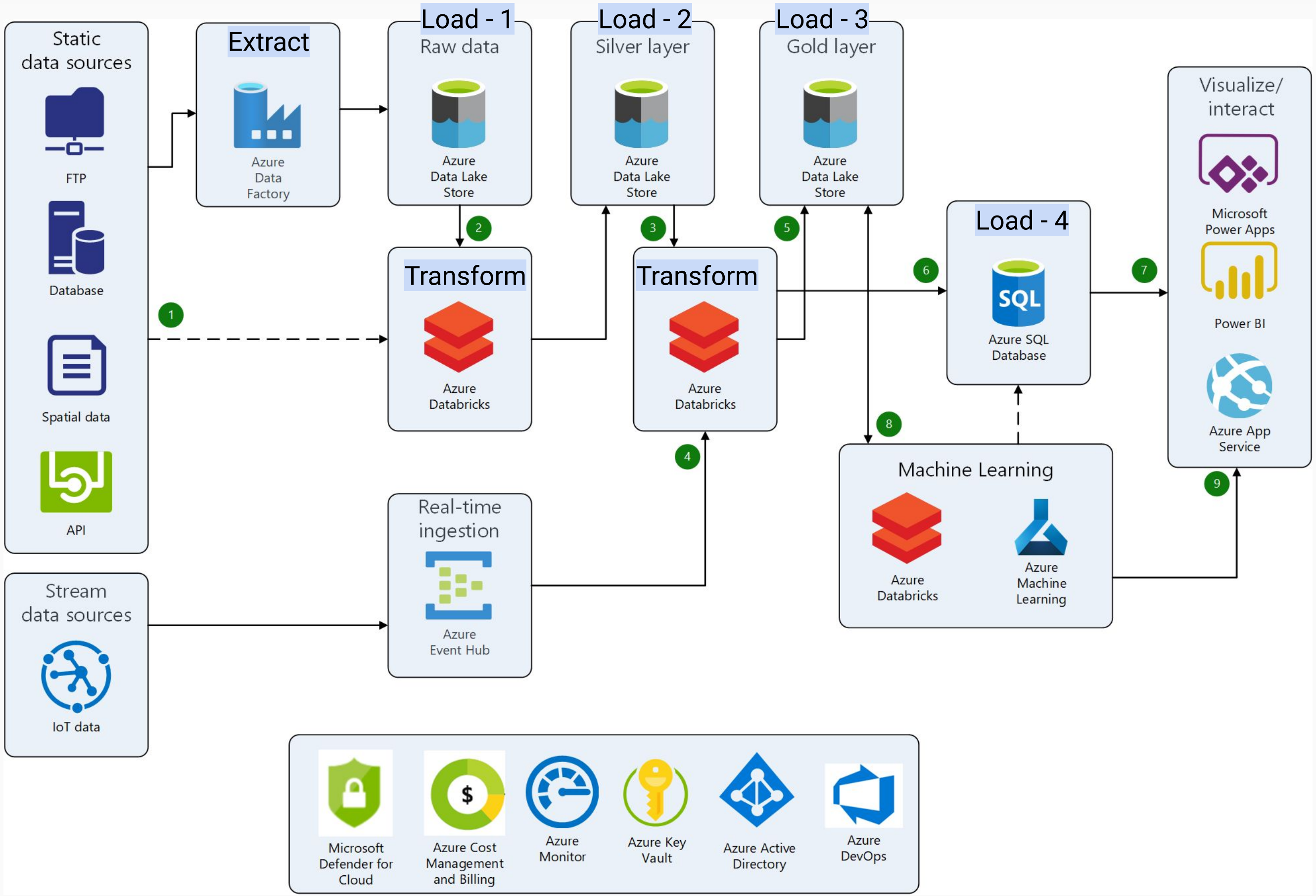
Concept III

ETL Steps / Tools

Concept IV

(OLTP Transactional Processing vs. (OLAP)Analytical Processing

Intro to Data Platforms II



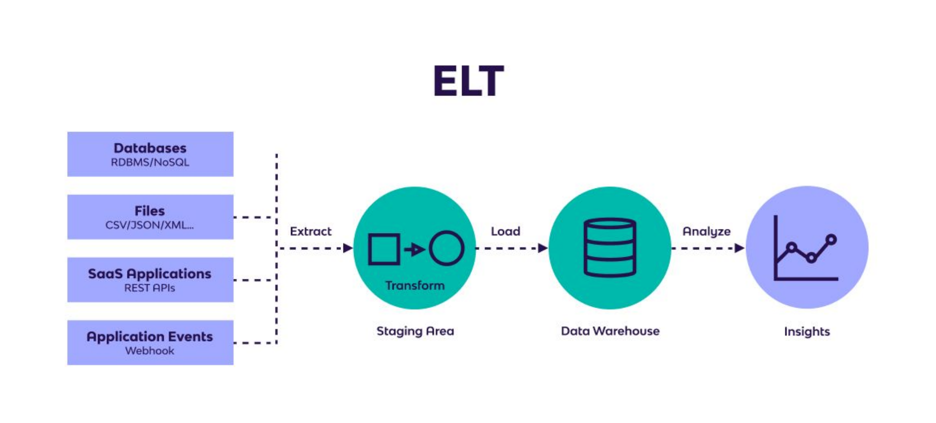
Concept V
Database vs. Datawarehouse

Concept VI
Big Data

Concept VII
Batch vs. Streaming

ELT | Extract Transform Load & ETL vs. ELT

- Extract
  - Collect raw data from one or more sources.
  - Raw data can be in various formats
  - Need to combine into a consistent format.
- Transform
  - Raw data is processed into a consistent format.
  - Cleaning, reformatting, addressing duplicates/missing values.
  - Typically done on a separate server.
- Load
  - Cleaned data is inserted into a target database, data store, or data warehouse.



ETL vs. ELT

Business Scenarios for ETL and ELT

ETL	ELT
Source and target databases are different (e.g., Oracle source and SAP target databases)	Source and target databases are same (e.g., Oracle source and target databases)
Data volume is small or moderate	Data volume is large
Data transformations are compute-intensive	Data transformations are less complex
Data is structured	Data is unstructured

ETL vs ELT

ELT is particularly useful for high-volume, unstructured datasets as loading can occur directly from the source. Ideal for big data management since it doesn’t need much upfront planning for data extraction and storage. ETL is more useful, for relational databases (of small size) and needs lot of upfront data modelling design.

Questions ?

Summary

Modern data has many forms

- Big/Small
- Batch/Stream
- Relational/Non-relational

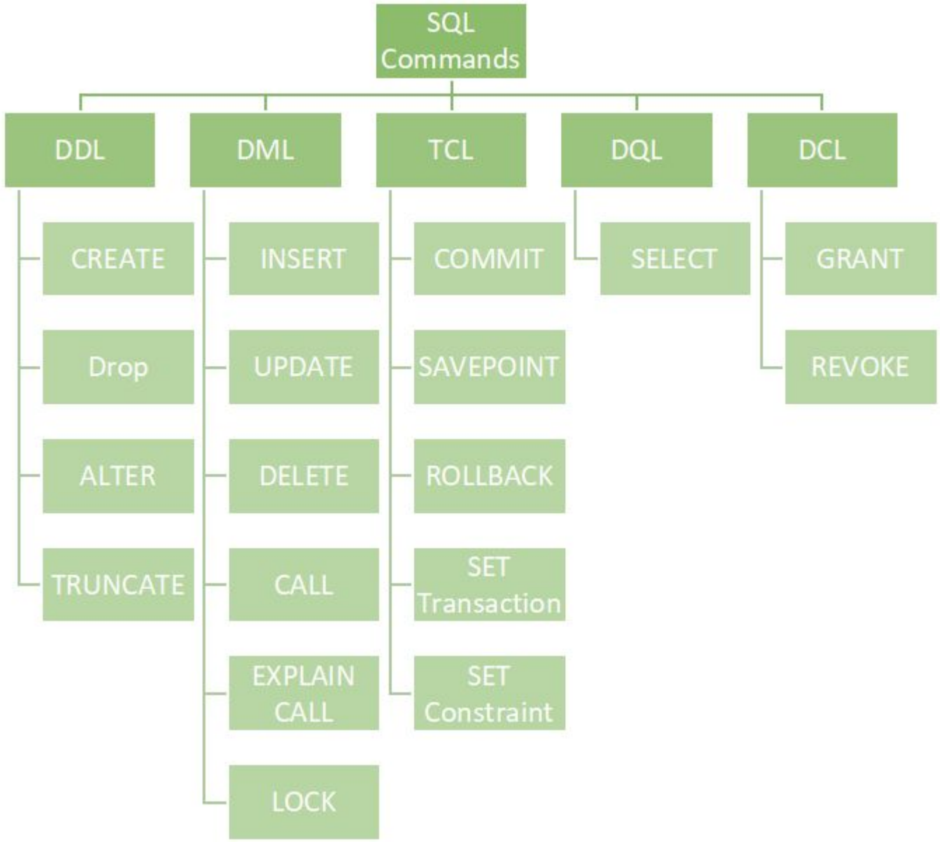
Why we need Data Platforms

- Data Stores (DB,Lake,DW)
- ETL Tools for pipelines
- BI/ML for serving

What is ETL ?

ETL vs. ELT

DDL/DML/DQL



SQL Commands
DDL
DML
TCL
DQL
DCL
CREATE
Drop
ALTER
TRUNCATE
INSERT
UPDATE
DELETE
CALL
EXPLAIN CALL
LOCK
COMMIT
SAVEPOINT
ROLLBACK
SET Transaction
SET Constraint
SELECT
GRANT
REVOKE

DDL Statements in SQL

- A table can be created using the ‘CREATE TABLE’ statement.
- A table can be deleted using the ‘DROP TABLE’ statement.
- A table can be modified using the ‘ALTER TABLE’ statement.
  - Add a column, drop a column, or change a column’s data type.
- There is also:
  - ‘TRUNCATE’: removes all records from a table
  - ‘COMMENT’: adds comments to the data dictionary
  - ‘RENAME’: renames an object.

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
   ....
);
```

```
DROP TABLE table_name;
```

```
ALTER TABLE table_name
ADD column_name datatype;
```

```
ALTER TABLE table_name
DROP COLUMN column_name;
```

```
ALTER TABLE table_name
MODIFY COLUMN column_name datatype;
```

```
Truncate TABLE newdb_lhl.student_copy
DROP TABLE newdb_lhl.student_copy
```

Data Manipulation Language (DML)

- DML : Store, modify, retrieve, delete, and update data/tables/databases.
- New rows in a tabl via. ‘INSERT INTO’ statement.
  - Two formats
  - Can insert multiple rows at once, separated by a comma.
  - Can insert for only a subset of columns.
- Insert from another table, use CTAS.
- Load from a file (CSV/Parquet)
  - Postgres syntax shown

```
CREATE DATABASE IF NOT EXISTS newdb_lhl
```

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

```
INSERT INTO table_name
VALUES (value1, value2, value3, ...);
```

```
INSERT INTO table2
SELECT * FROM table1
WHERE condition;
```

```
CREATE TABLE student_copy AS SELECT * FROM newdb_lhl.student;
```

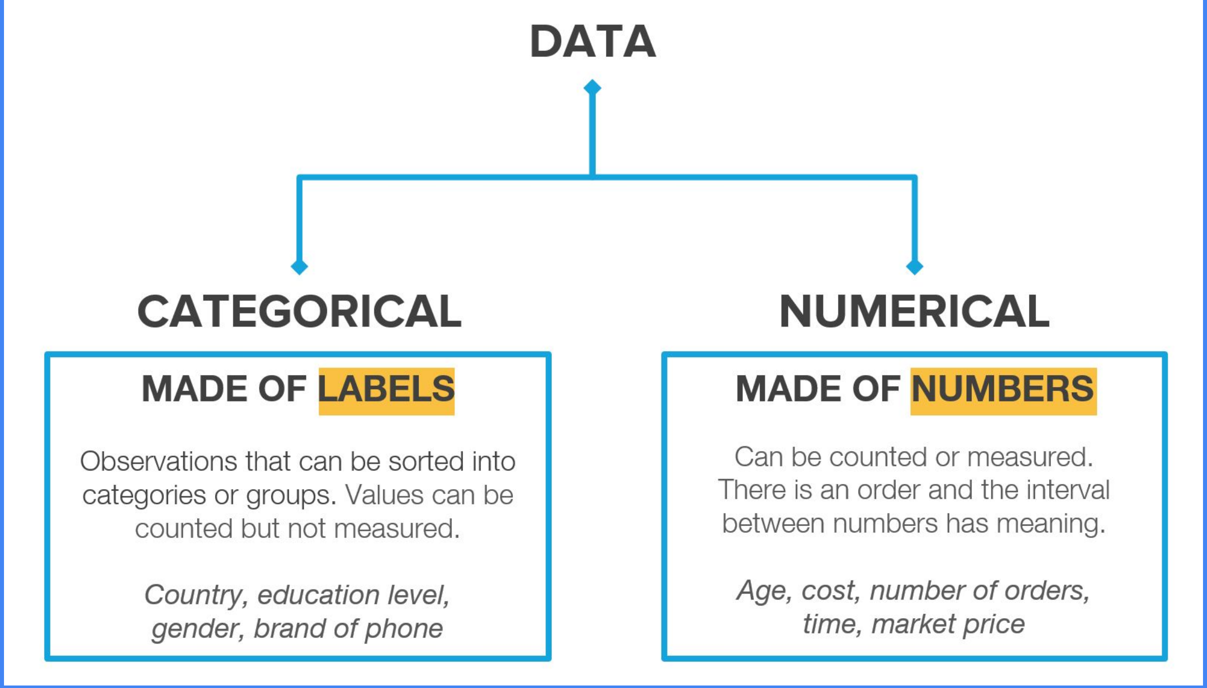
Summary

CREATE TABLE [table_name]
CREATE DATABASE [db_name]
DROP vs. TRUNCATE
INSERT INTO
CREATE TABLE AS SELECT (CTAS)

Questions ?

Analyzing Data
Part III

Data Types


DATA
CATEGORICAL
MADE OF LABELS
Observations that can be sorted into categories or groups. Values can be counted but not measured.
Country, education level, gender, brand of phone
NUMERICAL
MADE OF NUMBERS
Can be counted or measured. There is an order and the interval between numbers has meaning.
Age, cost, number of orders, time, market price

WHAT TYPES OF DATA DO WE HAVE?

Let's categorize the following examples:

Number of employees

Region

Age group

Time

WHAT TYPES OF DATA DO WE HAVE?

NUMERICAL

Number of employees

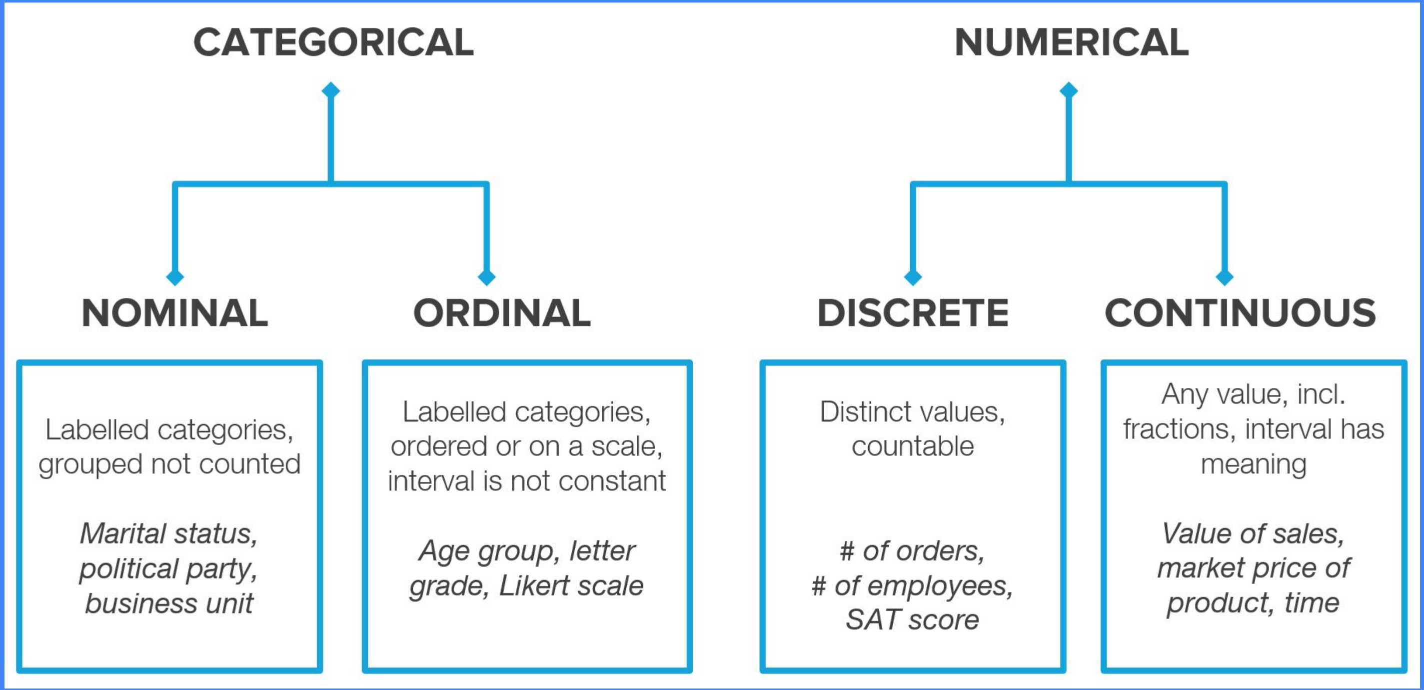
Time

CATEGORICAL

Region

Age group

Data Types


CATEGORICAL
NUMERICAL
NOMINAL
ORDINAL
DISCRETE
CONTINUOUS
Labelled categories, grouped not counted
Marital status, political party, business unit
Labelled categories, ordered or on a scale, interval is not constant
Age group, letter grade, Likert scale
Distinct values, countable
# of orders, # of employees, SAT score
Any value, incl. fractions, interval has meaning
Value of sales, market price of product, time

NOW WHAT TYPES OF DATA DO WE HAVE?

Let’s categorize the following examples:

NUMERICAL	CATEGORICAL
Number of employees	Region
Time	Age group

NOW WHAT TYPES OF DATA DO WE HAVE?

NOMINAL

Region

ORDINAL

Age group

DISCRETE

Number of employees

CONTINUOUS

Time

- Descriptive
    - Helps us understand **what** happened.
    - Interpretation of historical data to identify patterns.
    - Measures of central tendency, measures of variability, frequency distributions.
- Diagnostic
    - Helps us understand **why** something happened.
    - Identifying correlations between variables.
    - Determining factors that drive revenue, decrease turnover.
- Predictive
    - Attempts to answer what is **likely** to happen.
    - Uses past trends to forecast what might happen in the future.
    - Churn risk, sales forecasting, next best offers.
- Prescriptive
    - What do we need **to do**.
    - Uses optimization and simulation algorithms to advise on possible outcomes.
    - Machine learning, artificial intelligence.

- Measures of central tendency
  - Used to describe a typical value of the data.
  - Mean, median, mode.
- Measures of dispersion
  - Used to describe the spread of data.
  - Range, standard deviation, variance.
- Measures of Frequency
  - Used to describe the distribution of categorical data.
  - Counts, percentages, frequencies.
- Measures of Position
  - Used to describe the distribution of numerical data.
  - Ranks, Percentiles, quartiles.

- All of these can be calculated using SQL.

Summary

Categorical vs Numerical

Ordered Categorical : Ordinal e.g.Age_group
Group Categorical: Nominal (Color)

Discrete Numerical: No. of employees
Continuous Numerical: Time

Types of Analysis:
-Decriptive, Prescriptive, Predictive

Optional : Demo

SQL Descriptive analytics

SQL driven analytics

1

```
%sql
SELECT  MAX(salary) AS salary_max
      , MIN(salary) AS salary_min
      , CONCAT(MIN(salary),'-to-',MAX(salary)) AS salary_range
      , AVG(salary) AS salary_mean
      , percentile(salary, 0.25) AS quantile_1
      , percentile(salary, 0.5) AS quantile_2
      , percentile(salary, 0.75) AS quantile_3
      , std(salary) AS salary_std
      , variance(salary) AS salary_var
FROM newdb_lhl.people_10m
```

(2) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [salary_max: integer, salary_min: integer ... 7 more fields]

Table +

	salary_max	salary_min	salary_range	salary_mean	quantile_1	quantile_2	quantile_3	salary_std	salary_var
1	180841	-26884	-26884-to-180841	72633.0076033	59140	72638	86134	20003.229358500066	400129184.76875895

Cmd 8

2

```
%sql
SELECT
    people_10m.gender,
    ROUND(COUNT(salary)/100000,2) AS salary_count_in100K
FROM people_10m
GROUP BY people_10m.gender
ORDER BY people_10m.gender ASC
```

(2) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [gender: string, salary_count_in100K: double]

Table +

	gender	salary_count_in100K
1	F	51.87
2	M	48.13

Showing all 2 rows. | 1.16 seconds runtime

Mode

3

```
%sql
SELECT
    people_10m.salary AS salary_mode,
    COUNT(*) as count
FROM people_10m
GROUP BY people_10m.salary
ORDER BY COUNT(*) DESC
LIMIT 1
```

(2) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [sa

Table +

	salary_mode	count
1	72436	249

Questions ?

Thanks!