Module 2 - Lecture 3

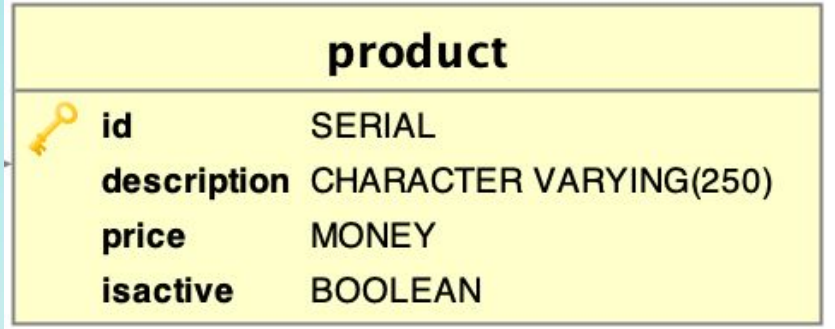
SQL Keys, Joins, and Unions

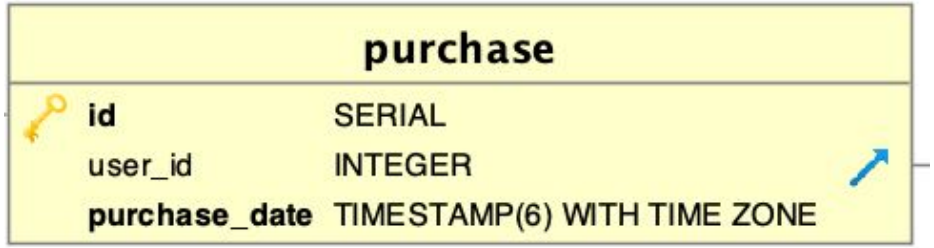
REVIEW

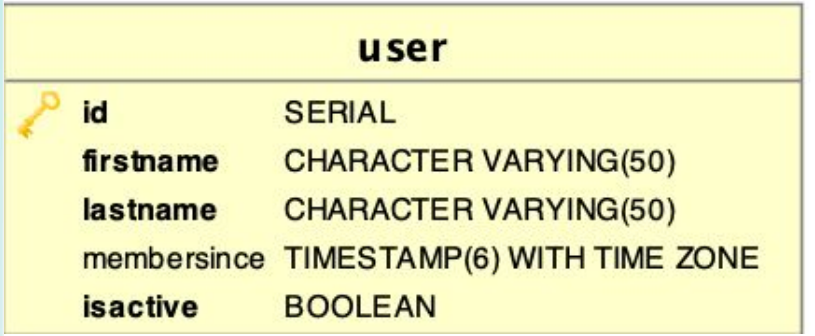
- How do you order query results?
- How do you limit how many results we get back?
- What are aggregate functions?
- How do you get records into summary rows?

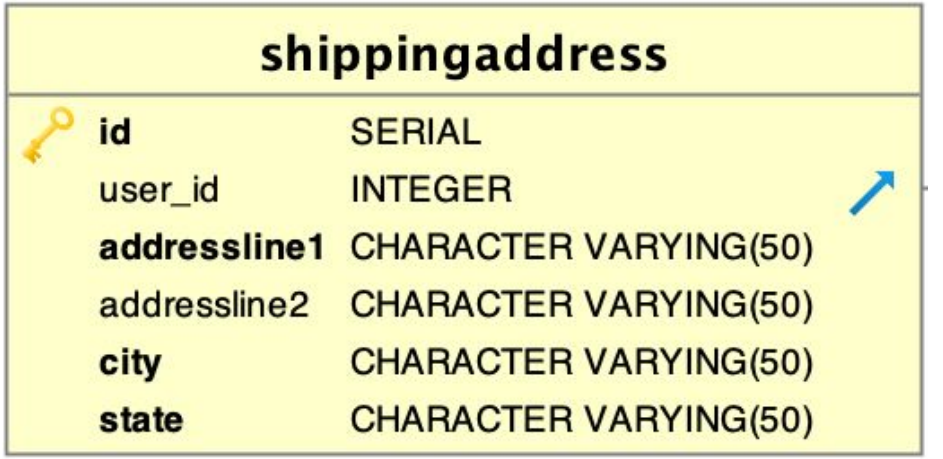
amazon.com

SCHEMA EXAMPLE



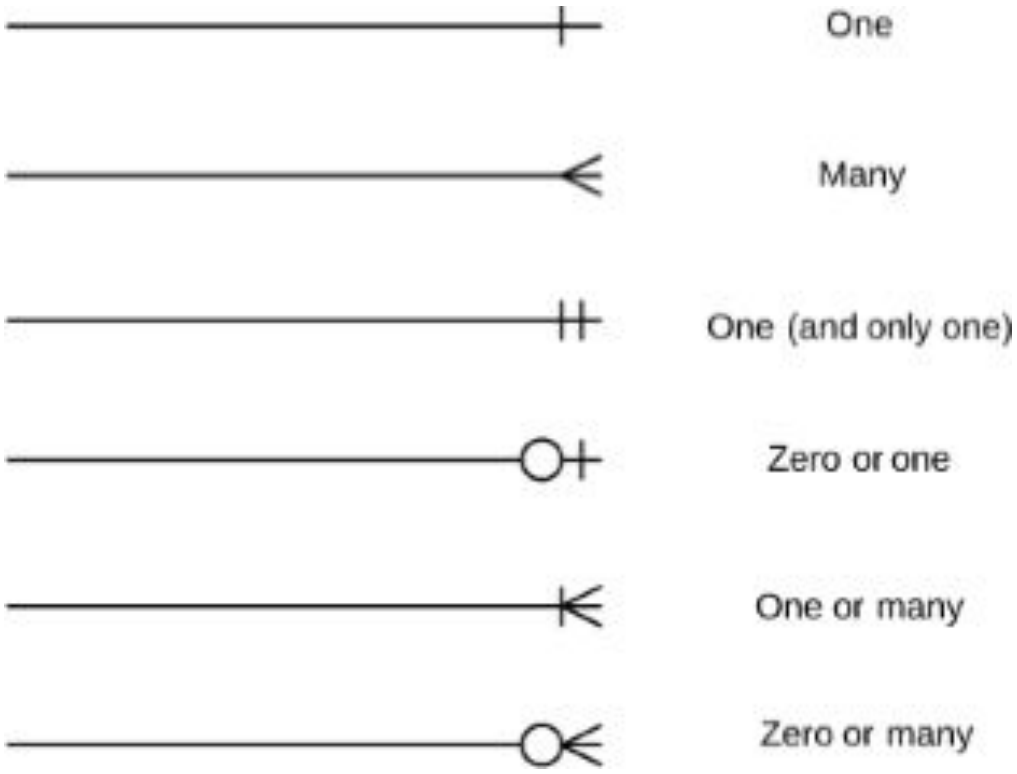







there must be an easier way of getting a drink than this!!
ICANHASCHEEZBURGER.COM

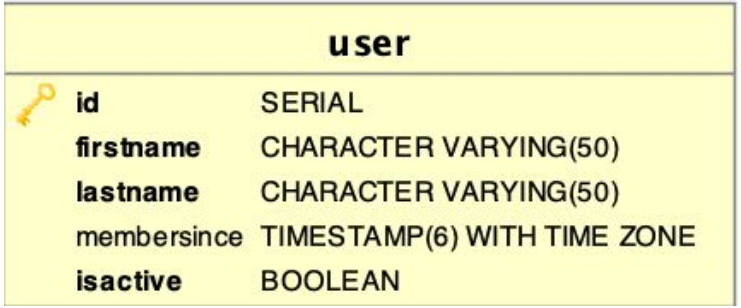
CARDINALITY AND ORDINALITY

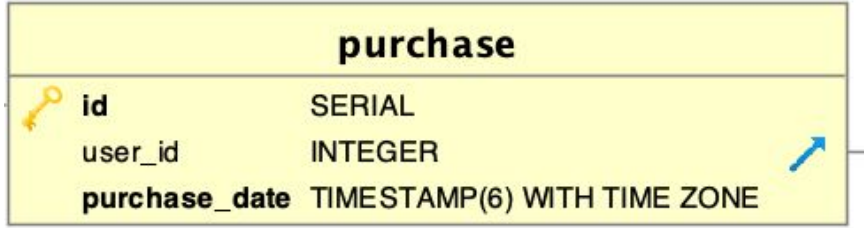


CARDINALITY AND ORDINALITY

- **Cardinality** refers to the maximum number of times that an instance in one entity can be associated with instances in a related entity.
- **Ordinality** refers to the minimum number of times it must be associated. E.g. mandatory or optional.

- Example: How many purchases can a given user have?





KEYS

Primary Keys uniquely identify a row in a table.

purchase	
id	SERIAL
user_id	INTEGER
purchase_date	TIMESTAMP(6) WITH TIME ZONE

user	
id	SERIAL
firstname	CHARACTER VARYING(50)
lastname	CHARACTER VARYING(50)
membersince	TIMESTAMP(6) WITH TIME ZONE
isactive	BOOLEAN

Foreign Keys are a field in a table that uniquely identifies a row in another table.

PRIMARY KEYS

- Are a type of constraint
- Must be unique
- Cannot be null
- May contain one or many columns
- Are considered to be natural or surrogate.
    - A surrogate key is synthetic. It is purely created as an identifier and has no relationship to the table. A common surrogate key is an integer that increments from 1 onward.
- Only one is allowed per table

PRIMARY KEY SYNTAX

As a “column constraint”

```
CREATE TABLE purchase
(
  id integer PRIMARY KEY
);
```

As a “table constraint”

```
CREATE TABLE purchase
(
  id integer,
  CONSTRAINT pk_purchase_id PRIMARY KEY (column1)
);
```

FOREIGN KEYS

- Are another type of constraint.
- May contain one or many columns
- The data type of the foreign key column must match the data type of the column it references.
- Can have more than one foreign key per table.
- Must reference a primary or unique key in another table.
  - Maintains **referential integrity** between two related tables.

FOREIGN KEY SYNTAX

As a “column constraint”

```
CREATE TABLE purchase
(
  id integer PRIMARY KEY
 user_id integer REFERENCES “user” (id)
);
```

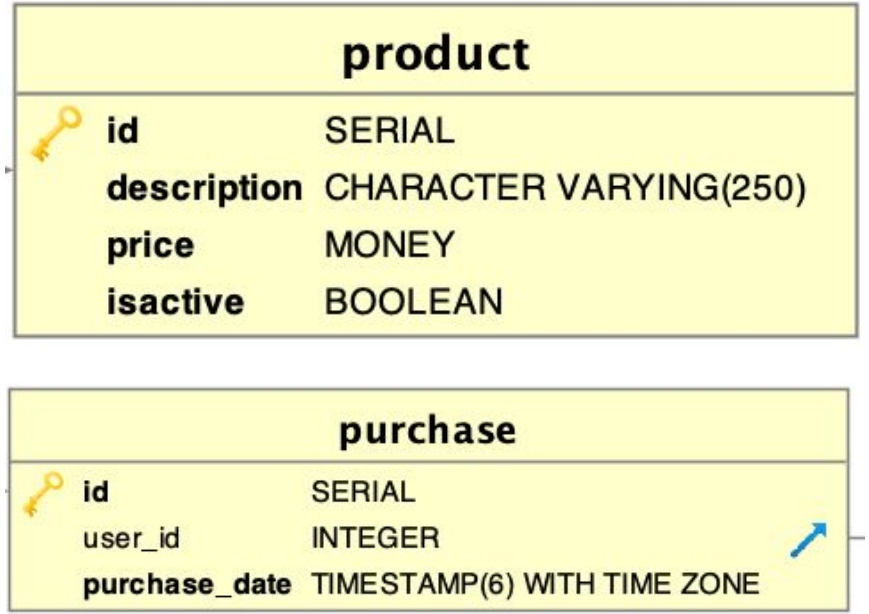
FOREIGN KEY SYNTAX

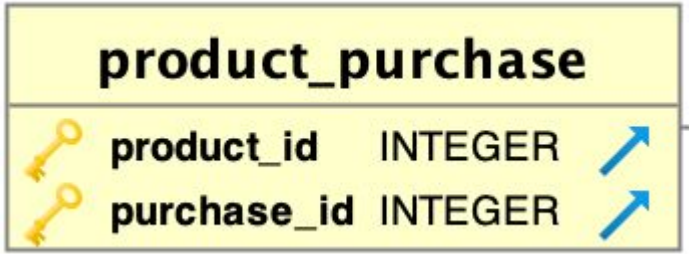
As a “table constraint”

```
CREATE TABLE purchase
(
  id integer PRIMARY KEY,
  user_id integer,
  CONSTRAINT fk_user_id FOREIGN KEY (user_id)
REFERENCES user (id)
);
```

CARDINALITY (revisited)

- How many products can be included in a purchase?
- How many purchases can include a product?





JOINS

combine columns

JOINS

SQL JOINs allow us to create queries that produce data from one or more tables.

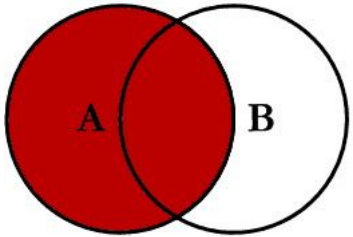
Recall the examples from earlier where we tried to find all purchases that John Henry has made and all products in those purchases?

- Let's rewrite those using JOINs.

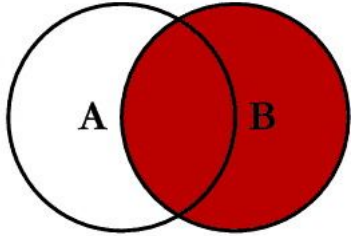
SYNTAX

```
 SELECT    table1.column, table2.column
 FROM      table1
[INNER JOIN | LEFT JOIN | RIGHT JOIN] table2
    ON table1.column = table2.column;
```

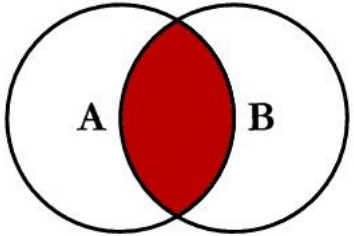
SQL JOINS



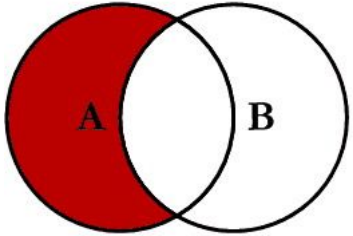

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



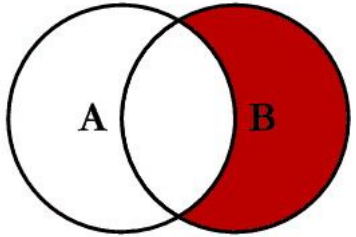

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



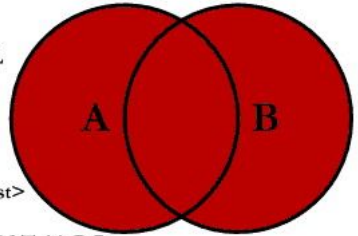

```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



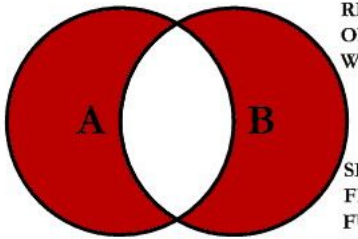

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```




```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```


UNIONS

combine rows

UNIONS

SQL UNIONs combine the results of two or more queries into a single result set.

- The number of columns involved must match exactly and the data types must be identical.
- The data types must be compatible with each other.
- The names of the columns do not need to match.
- Duplicate rows are removed by default. They can be included using UNION ALL.

SYNTAX

```
SELECT table1.column FROM table1 [WHERE] [...]
UNION [ALL]
SELECT table2.column FROM table2 [WHERE] [...];
```

INDEXES

INDEXES

Indexes are a common way to enhance database performance.

- Reorganize how the data is stored.
  - Clustering will actually reorder the data on disk.
  - Non-clustered indexes will have a reference to the data's location.

- Helps to retrieve rows much faster.
  - They also add overhead.

- Primary keys automatically add an index.

- Can contain multiple columns.

QUESTIONS?