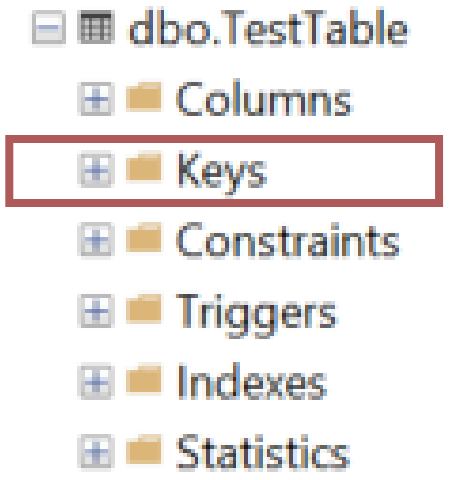
Section 7: Joining Tables

- *Primary Key Review*
- *Relational Databases*
- *Data Relationships*
- *Foreign Keys*
- *Setting up Primary and Foreign Keys*
- *Inner Join*
- *Left Joins*
- *Right Joins*
- *Review Cheat Sheet*

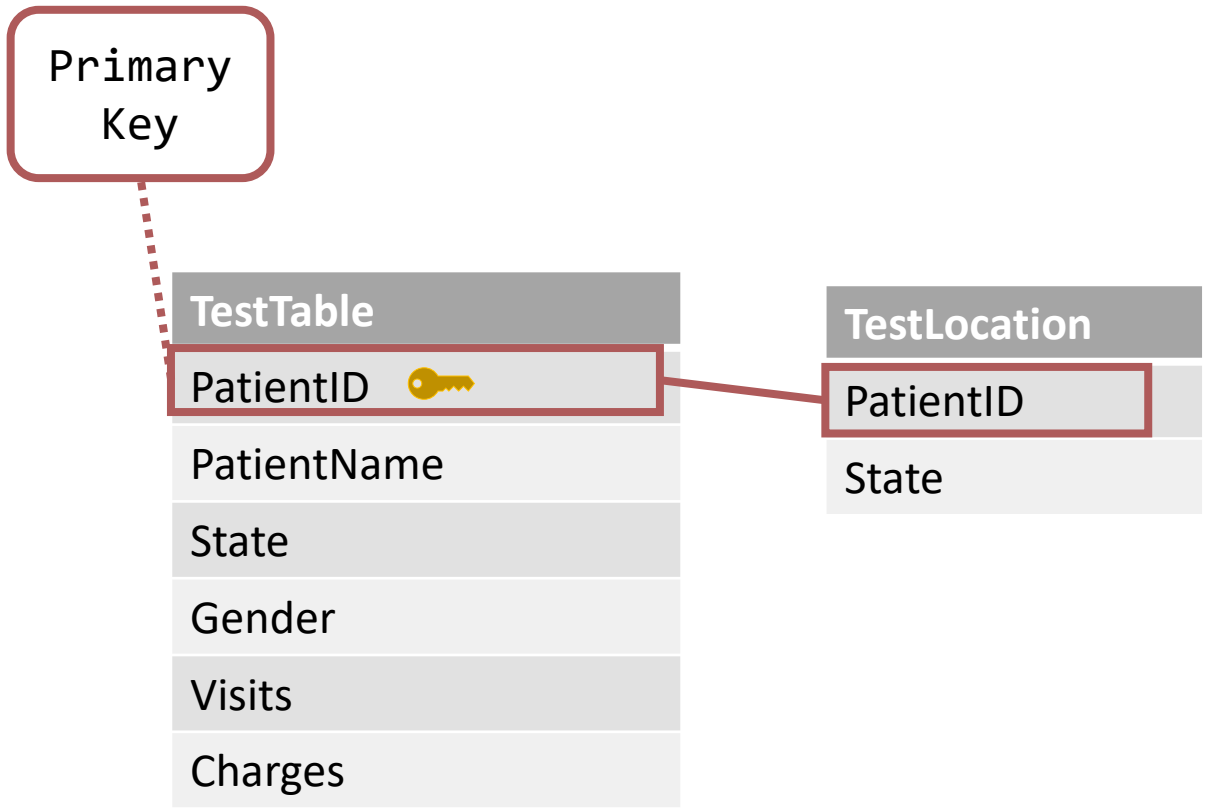
Database example



PatientID	PatientName	PatientSta	Gender	Visits	Charges
12355	Fred	CA	M	3	500
12355	Fred	CA	M	3	500
12355	Fred	CA	M	3	500
12355	Fred	CA	M	3	500
12355	Fred	CA	M	3	500
12355	Fred	CA	M	3	500

-> Example Statement

Section 7: Primary Key Review



PatientID	PatientName	Gender	Visits	Charges
12345	John	M	3	$ 200
12346	Jane	F	1	$ 400
12347	Alex	F	6	$ 900
12348	Bob	M	7	$ 8,000
12349	Josh	M	12	$ 19,000
12350	Stephanie	F	18	$ 25,000
12351	Amber	F	4	$ 400
12352	Brittany	F	6	$ 4,000
12353	Bill	M	8	$ 5,000
12354	Nate	M	22	$ 28,000

PatientID	State
12345	AL
12346	AK
12347	AZ
12348	CA
12349	CO
12350	FL
12351	GA
12352	GA
12353	UT
12354	WY

-> Example Statement

PatientID	PatientName	State	Gender	Visits	Charges
12345	John	AL	M	3	$ 200

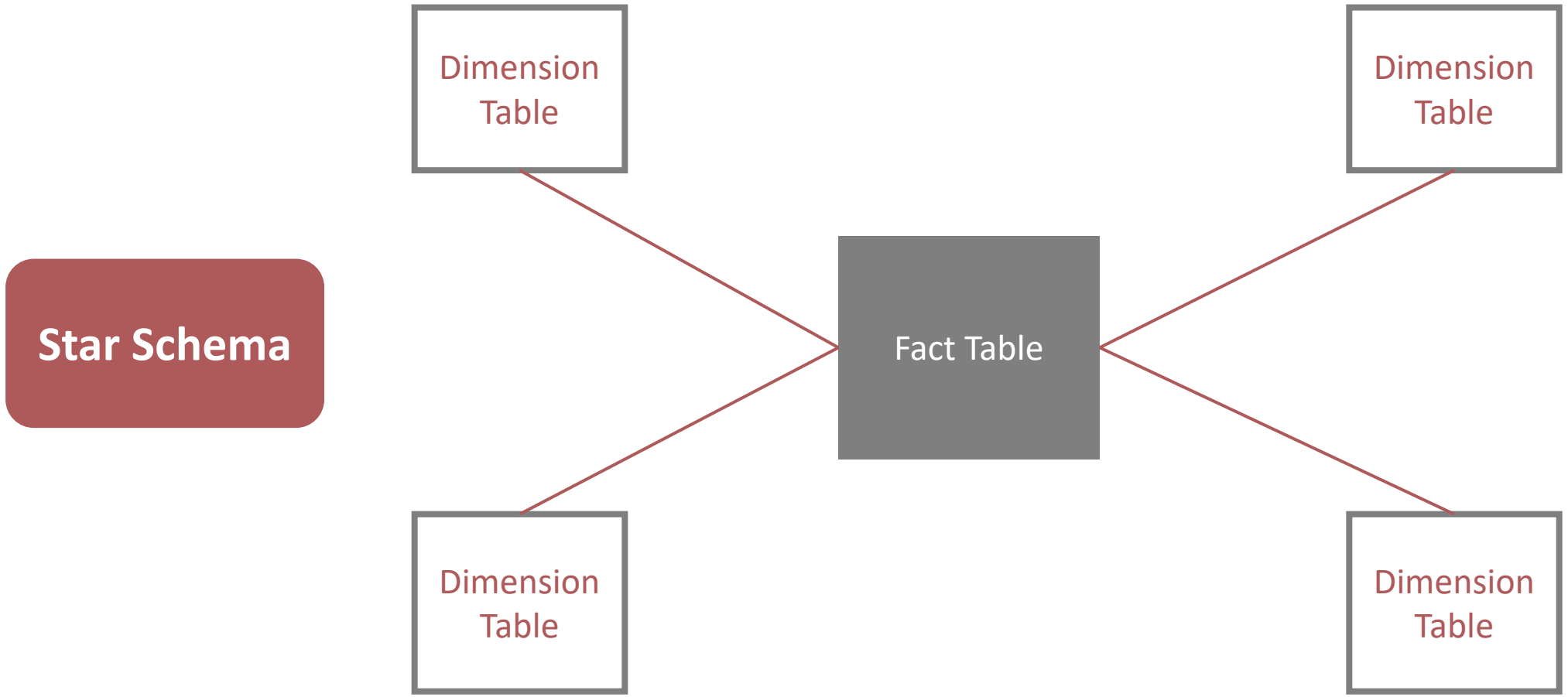
Section 7: Primary Key Review

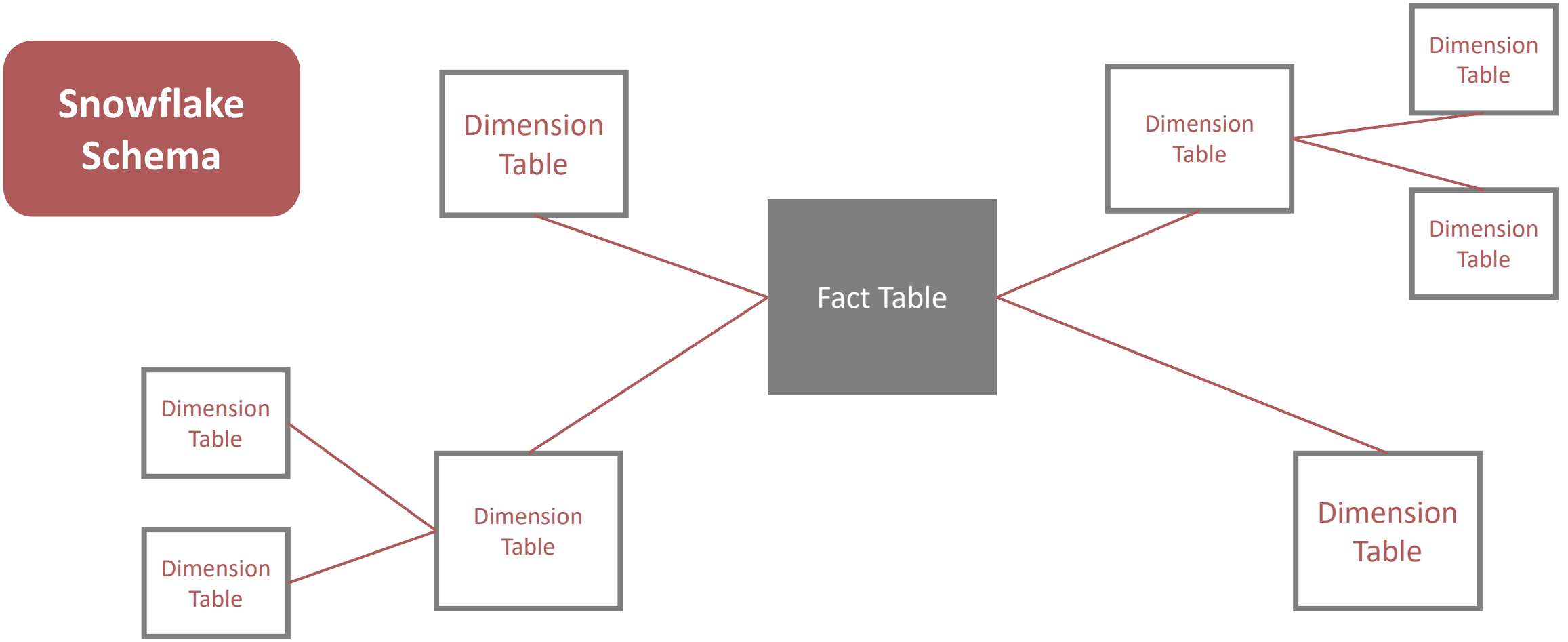
Primary Key Reminders

1. A Primary key must contain unique values
2. A table can only have one primary key
3. The primary key identifies each record in a table and can connect multiple tables together

-> Example Statement

```
-> CREATE TABLE TestTable
(PatientID int NOT NULL PRIMARY KEY,
PatientName varchar(255) NULL,
PatientState varchar(255) NULL,
Gender varchar(255) NULL,
Visits int NULL,
Charges int NULL Default 0)
```


Star Schema
Dimension Table
Dimension Table
Fact Table
Dimension Table
Dimension Table


Snowflake Schema
Dimension Table
Dimension Table
Dimension Table
Dimension Table
Fact Table
Dimension Table
Dimension Table
Dimension Table
Dimension Table

Why can’t we just store all the data in a single table?

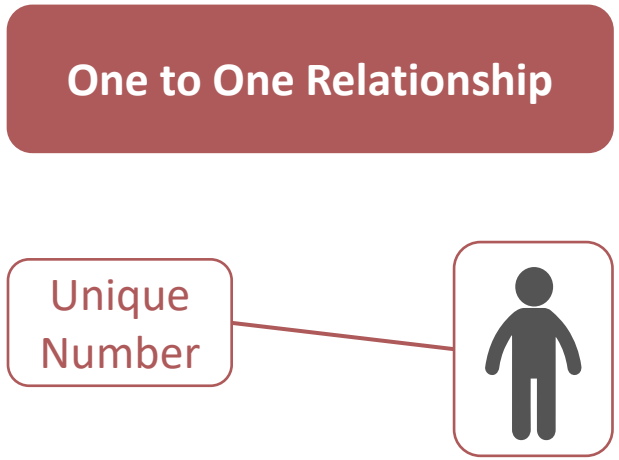
Section 7: Relational Databases

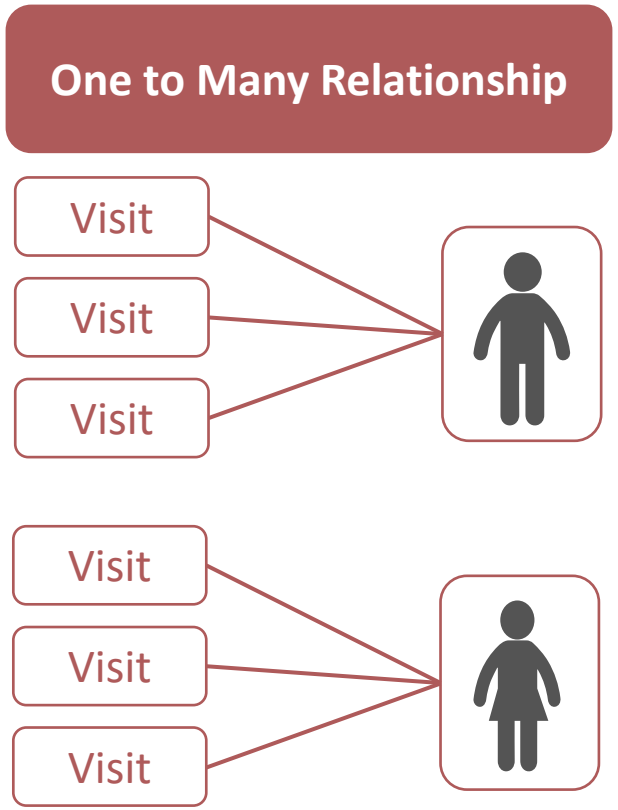
Patient Number	Visit ID	Patient Name	CPTCode	Date of Service	Location
12345	6789	Fred	99222	Jan. 1	West Clinic
12345	6789	Fred	90674	Jan. 1	West Clinic
12345	6790	Fred	99222	Jan. 5	West Hospital
12345	6791	Fred	99222	Jan. 6	East Hospital
12345	6791	Fred	96360 (IV)	Jan. 6	East Hospital
12345	6791	Fred	99222	Jan. 6	East Hospital

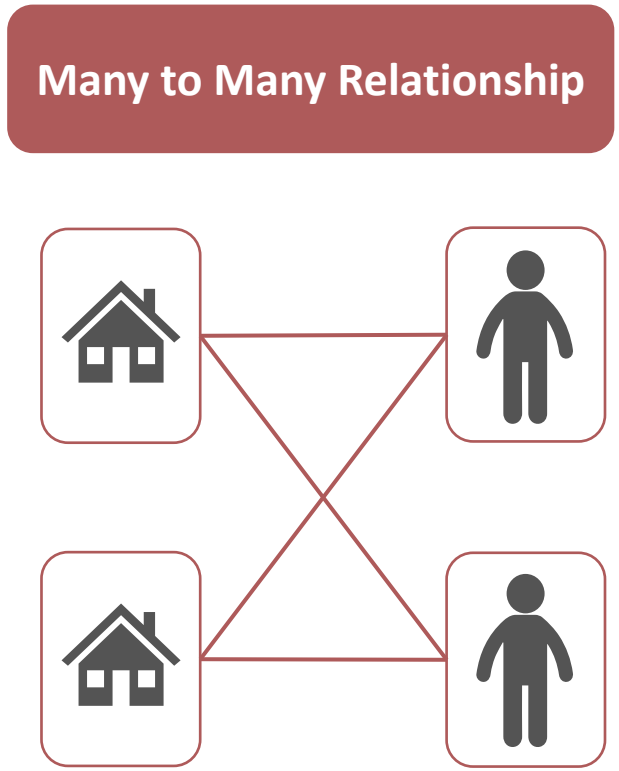
VS.

Patient Number	Visit ID	Date of Service	Location
12345	6789	Jan. 1	West Hospital
12345	6790	Jan. 5	West Clinic
12345	6791	Jan. 6	East Hospital

Visit ID	CPTCodes
6789	99222
6789	90674
6790	99222
6791	99222
6791	96360
6791	99222


One to One Relationship
Unique Number
Unique Number


One to Many Relationship
Visit
Visit
Visit
Visit
Visit
Visit


Many to Many Relationship

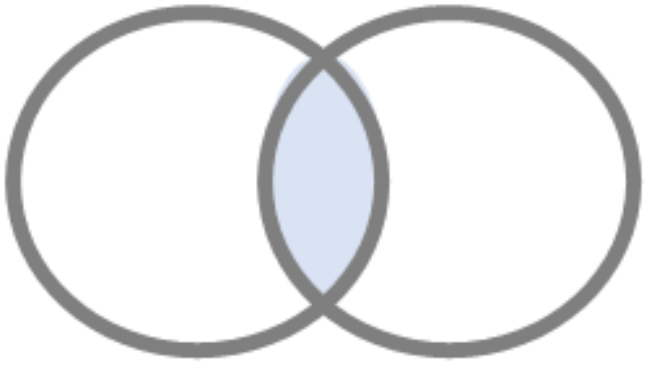


Table 1 Table 2

This is “left table”

```
Select *
   from TableName1
   INNER JOIN TableName2 on
     TableName1.Column = TableName2.Column


-> Select *
   from TestTable
   INNER JOIN HospitalTable on
     TestTable.LocationID = HospitalTable.LocationID
```

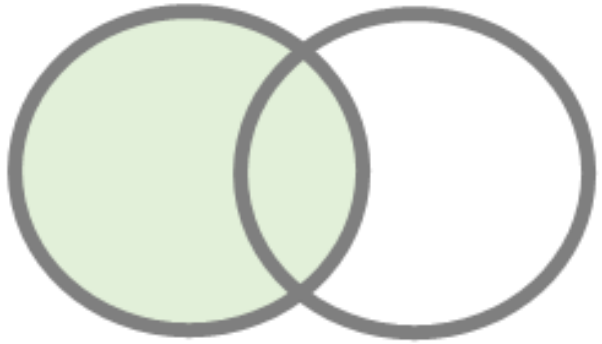


Table 1 Table 2

This is “left table”

```
Select *
   from TableName1
   LEFT JOIN TableName2 on
     TableName1.Column = TableName2.Column


-> Select *
   from TestTable
   LEFT JOIN HospitalTable on
     TestTable.LocationID = HospitalTable.LocationID
```

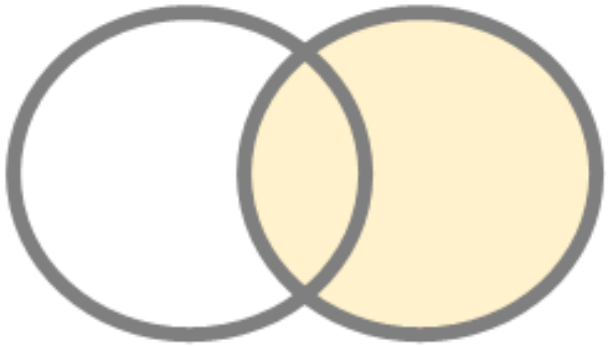


Table 1 Table 2

This is “left table”

```
Select *
   from TableName1
   RIGTH JOIN TableName2 on
     TableName1.Column = TableName2.Column


-> Select *
   from TestTable
   RIGHT JOIN HospitalTable on
     TestTable.LocationID = HospitalTable.LocationID
```

Section 7: Review Cheat Sheet

Basic Syntax

```
SELECT * FROM table_name
  -> Populates the whole table
SELECT column1, col2, col3… FROM table_name
  -> Populates specified columns
WHERE col2 = condition (=,>,<,>=,<=)
  -> filter rows where column values meet
condition
GROUP BY col1, col3
  -> Groups rows that have the same values
HAVING Count(*) > value
  -> Limit Aggregated Data
ORDER BY col4 (DESC or ASC)
  -> Order you results by a column
```

Useful Keywords when using SELECTS

```
DISTINCT -> Returns unique rows
BETWEEN a AND b -> Limits range of values
LIKE -> Pattern Search within the column values
IN(a,b,c) -> Returns values contained among list
TOP 100 -> Select top number of rows
```

Aggregation Functions

```
COUNT -> Count of rows SUM -> Cumulates values
AVG -> Avg's Values Max/Min -> Small/large values
```

Table Manipulation

```
CREATE TABLE table_name (col1 datatype, col2 datatype…)
  -> Creates new table, specify the type of data in columns
DROP TABLE table_name
  -> Permanently deletes data table
TRUNCATE TABLE table_name
  -> Deletes data values in table, but table still exists
INSERT INTO table_name (col1, col2) VALUES (value1, value2)
  -> Insert data into created table
ALTER TABLE table_name ADD column_name datatype
  -> Add or Delete columns from table
UPDATE TABLE table_name SET col1 = value1, col2 = value2…
  -> Update existing records in a table
```

Joins

```
SELECT * FROM table1_name INNER JOIN table2_name ON
     table2_name.column1 = table1_name.column1
  -> Joining two tables using like columns
INNER JOIN -> Combing rows from tables where JOIN is true
LEFT JOIN -> Returns all records from left table and
matched records from the right table
RIGHT JOIN -> Returns all records from right table and
matched records from the left table
```

Joins Visualized