

# Retrieving data from multiple tables



# Basic Terminologies

- a **primary key** is a single field or combination of fields that uniquely defines a record. None of the fields that are part of the primary key can contain a null value. A table can have only one primary key.
- A **foreign key** is a constraint that's added to a table. It allows you to specify that a column in a table refers to the primary key of another table.
  - It's used to relate data in two tables and improve the integrity of your data.
  - the concept of *referential integrity* is all about maintaining and enforcing this parent-child relationship.

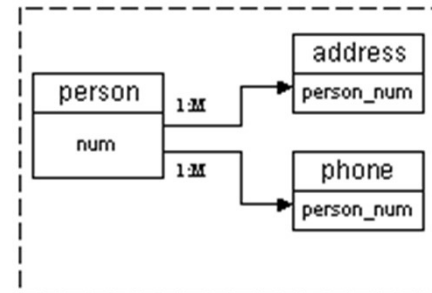
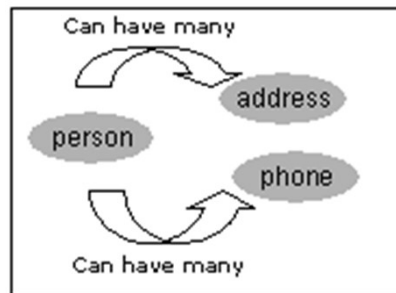
# Basic Terminologies

- **Logical table relationship**

- The relationship between two tables can be expressed in meaningful terms

- **Physical table relationship**

- The physical diagram depicts the primary key and foreign key relationship between our entities. Our primary table is "person". The foreign key column is the column "person\_num" in both the address and the phone tables respectively.



# Basic Terminologies

- A constraint is used to define **integrity in database** -a rule that restricts the values in a database. There are six types of integrity constraints.
  - A NOT NULL constraint prohibits a database value from being null.
  - A unique constraint prohibits multiple rows from having the same value in the same column or combination of columns but allows some values to be null.
  - A primary key constraint combines a NOT NULL constraint and a unique constraint in a single declaration. That is, it prohibits multiple rows from having the same value in the same column or combination of columns and prohibits values from being null.
  - A foreign key constraint requires values in one table to match values in another table.
  - A check constraint requires a value in the database to comply with a specified condition.
  - A REF column by definition references an object in another object type or in a relational table. A REF constraint lets you further describe the relationship between the REF column and the object it references.

# Cardinal Relationships: One to One

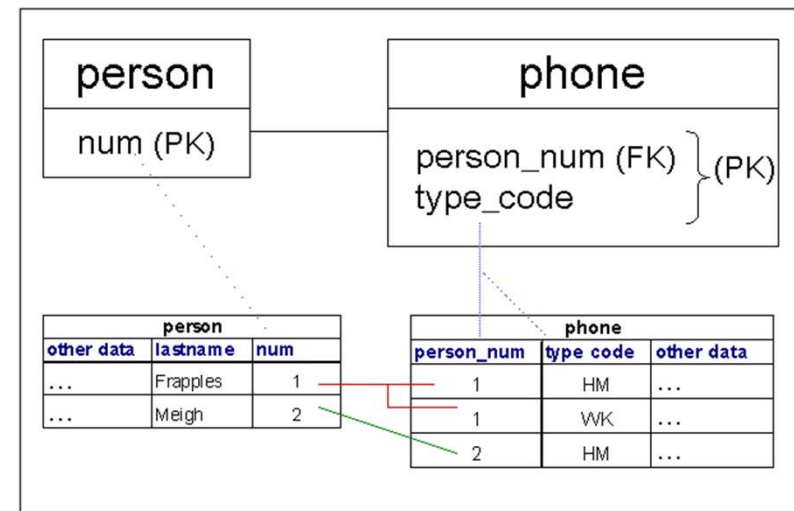
- **Primary Key (PK) - Foreign Key (FK) LAW: *If the foreign key column represents the entire primary key (or the entire index) then the relationship between the tables will be one to one (1:1).***
- What would we need to do to create a one to one relationship between the person table and the address table?
  - If you answered change the primary key of the address table to be only the column person\_num that is correct.
- But what would be a key drawback of doing this?
  - If you answered we would only be able to store one address per person you would be correct as well.

# Cardinal Relationships: many to one

- Understanding Cardinal Relationships
  - Knowing how foreign key constraints work allow us to understand how tables relate to one another.
  - As long as we can determine the primary key of each of our tables, and what the foreign key relationship is we can understand the relationship between two tables (one to many, one to one)
  - We can begin to understand and appreciate the difficulties and many challenges faced in designing relational databases. e)

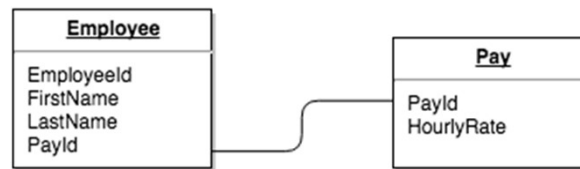
# Cardinal Relationships: many to one

- Primary Key (PK) person table: num
- **Primary Key (PK) - Foreign Key (FK) LAW: *If the foreign key column is part of the primary key (or part of an index) then the relationship between the tables will be one to many (1:M).***
- Primary Key (PK) phone table:
  - person\_num , type\_code
- Foreign Key relationship between person and phone:
  - person.num = phone.person\_num



# Cardinal Relationships: One to One

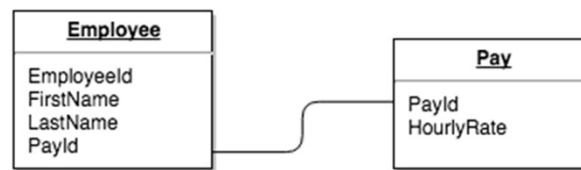
- A row in table A can have only one matching row in table B, and vice versa.
  - This is not a common relationship type, as the data stored in table B could just have easily been stored in table A.



- A one-to-one relationship can be used for security purposes, to divide a large table, and various other specific purposes.



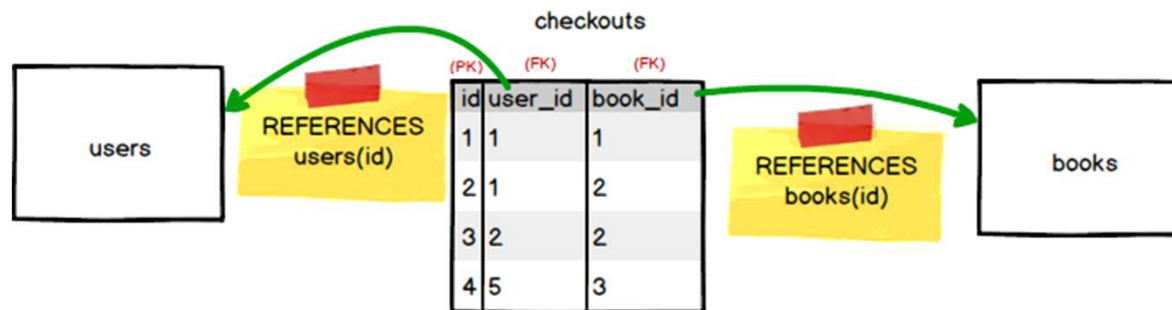
# Cardinal Relationships: One to One



- In this example, we could just as easily have put an HourlyRate field straight into the Employee table and not bothered with the Pay table.
- However, hourly rate could be sensitive data that only certain database users should see.
- So, by putting the hourly rate into a separate table, we can provide extra security around the Pay table so that only certain users can access the data in that table.

# Cardinal Relationships: many to many

- A many-to-many relationship exists between two entities if for one entity instance there may be multiple records in the other table, and vice versa.
  - Example: A user can check out many books. A book can be checked out by many users (over time).
- For this sort of relationship we introduce a cross-reference, table.
  - This table holds the relationship between the two entities, by having two FOREIGN KEYS, each of which references the PRIMARY KEY of one of the tables for which we want to create this relationship.
  - We already have our books and users tables, so we just need to create the cross-reference table: checkouts.



# Retrieving data from multiple tables

- Questions