kokchun giang

using different **keys** in database to identify rows, ensure data integrity and enforce relationships between tables

identifying **primary key** in this pokemon table

unique identifier
for the row

pokedex	name
1	Bulbasaur
4	Charmander
7	Squirtle
25	Pikachu
39	Jigglypuff
52	Meowth
95	Onix
131	Lapras
150	Mewtwo
151	Mew

name might also be considered unique here, but some newer versions of pokemon there exists regional pokemons

pokedex is very stable and don't change over time hence it is chosen as a **primary key**

a primary key is a column or a combination of columns to **uniquely identify each row**

candidate keys are set of keys from which primary key can be chosen

pokedex	name
1	Bulbasaur
4	Charmander
7	Squirtle
25	Pikachu
39	Jigglypuff
52	Meowth
95	Onix
131	Lapras
150	Mewtwo
151	Mew

if there wasn't any regional pokemons, then both name and pokedex would be considered as **candidate keys**

candidate keys must have **uniqueness** over time

candidate keys that are not chosen as primary key is called **alternate keys**

natural and **surrogate keys**

pokedex	name
1	Bulbasaur
4	Charmander
7	Squirtle
25	Pikachu
39	Jigglypuff
52	Meowth
95	Onix
131	Lapras
150	Mewtwo
151	Mew

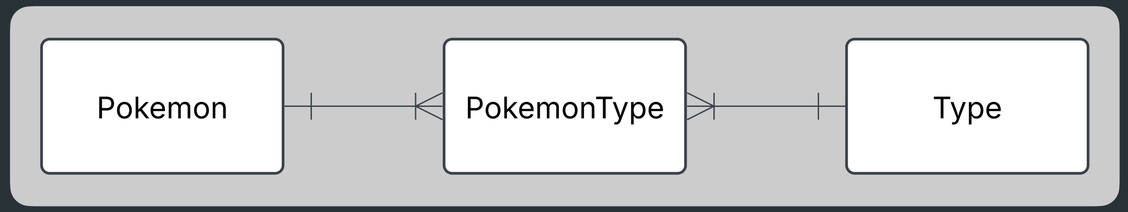
pokedex can be considered a **natural key** as it is a unique key that exists outside of the database

type_id	type_name
1	Grass
2	Poison
3	Fire
4	Water
5	Electric
6	Normal
7	Fairy

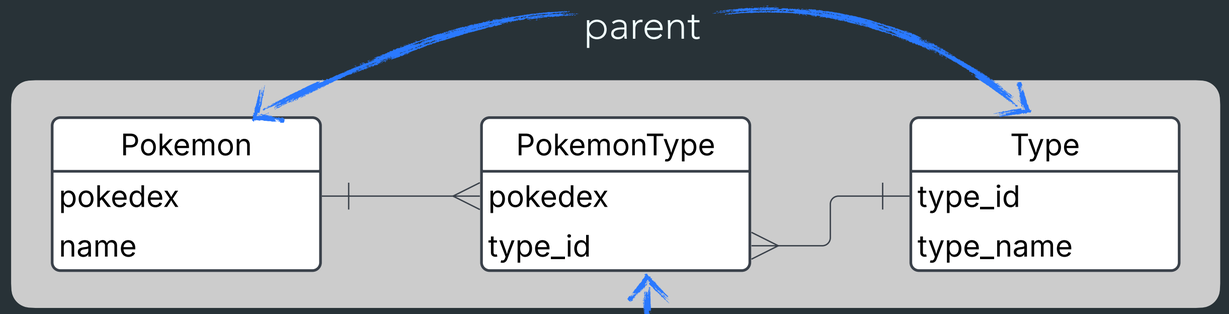
type_id is a **surrogate key** as it has no meaning outside the database

surrogate keys can also be created based on other attributes using hashing, more on that when we come to data warehouse course

different **data models** for the pokemon example



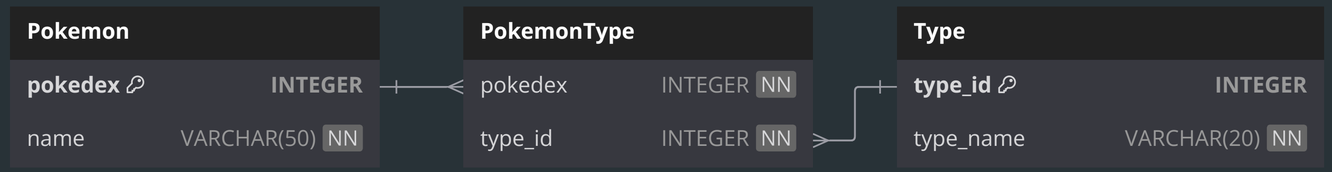
conceptual data model (CDM)



logical data model (LDM)

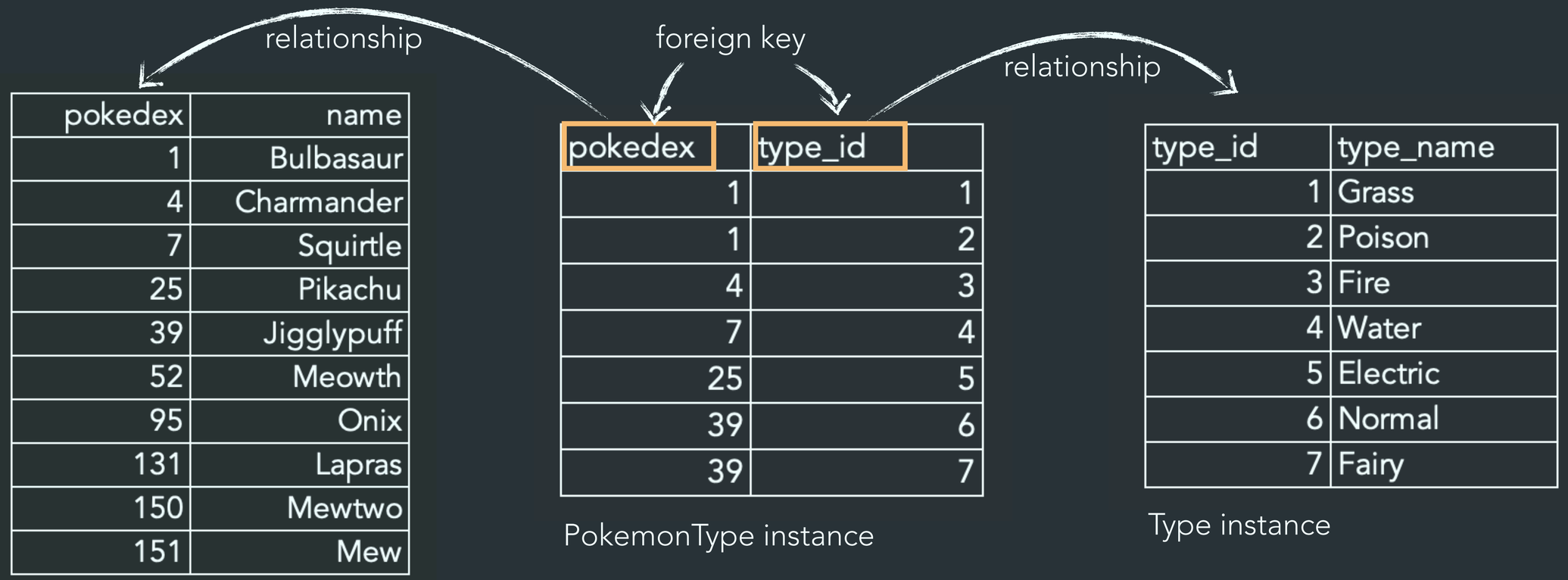
child entity depends on parent

what are the **differences** between these data models?



physical data model (PDM)

foreign key establish relationships between relations



pokedex	name
1	Bulbasaur
4	Charmander
7	Squirtle
25	Pikachu
39	Jigglypuff
52	Meowth
95	Onix
131	Lapras
150	Mewtwo
151	Mew

Pokemon instance

pokedex	type_id
1	1
1	2
4	3
7	4
25	5
39	6
39	7

PokemonType instance

type_id	type_name
1	Grass
2	Poison
3	Fire
4	Water
5	Electric
6	Normal
7	Fairy

Type instance

foreign keys appear in the **many sides** of the entities

foreign keys enforces referential integrity, which prevents orphaned records

foreign key enforces referential integrity

pokedex	name
1	Bulbasaur
4	Charmander
7	Squirtle
25	Pikachu
39	Jigglypuff
52	Meowth
95	Onix
131	Lapras
150	Mewtwo
151	Mew

Pokemon instance

pokedex	type_id
1	1
1	2
4	3
7	4
25	5
39	6
39	7

PokemonType instance

this means the database **restricts insertion** of a row with pokedex or type_id that don't exist in their parent table

type_id	type_name
1	Grass
2	Poison
3	Fire
4	Water
5	Electric
6	Normal
7	Fairy

Type instance

normal deletion of a row in a parent table that has related entry in child table is **not allowed**

ON DELETE CASCADE automatically **deletes dependent** rows on child table

foreign key that references own table

consider the following relation

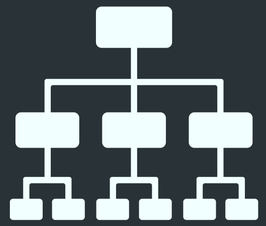
Employee(employee_id, first_name, last_name, manager_id)

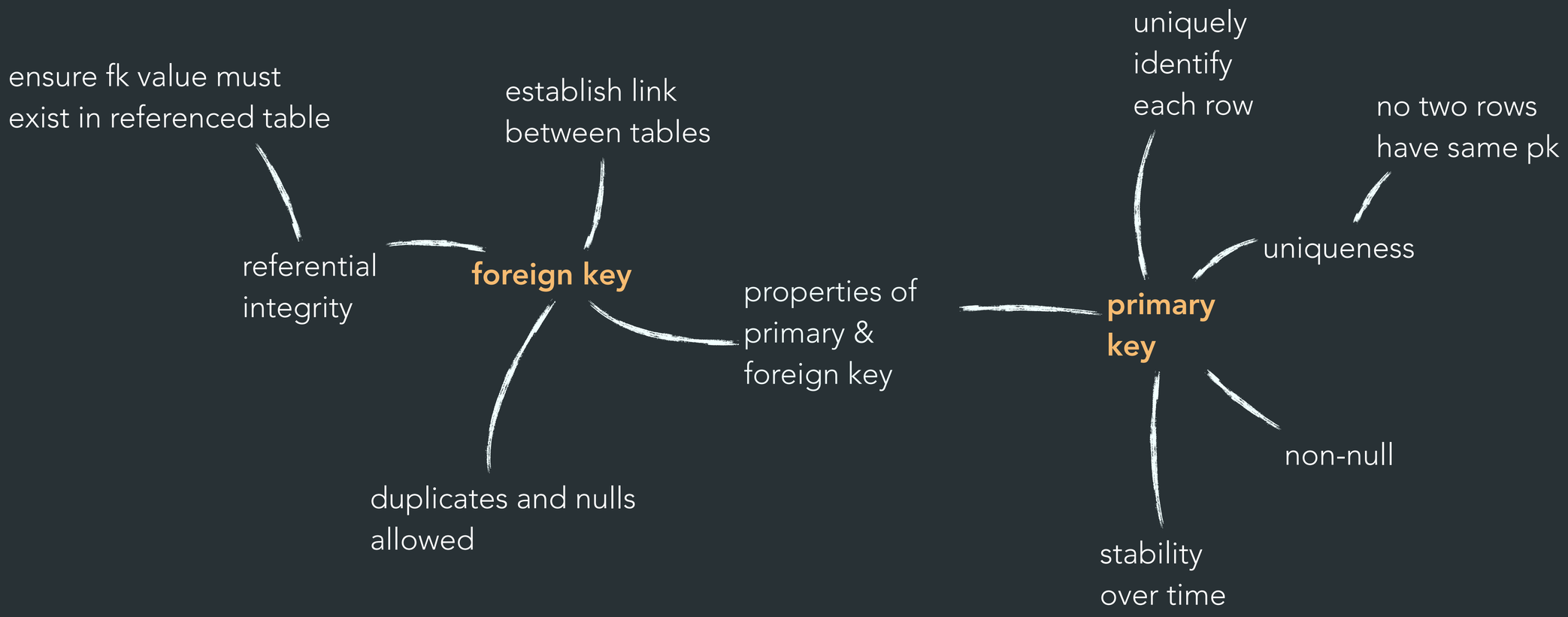
primary key

foreign key that references employee_id in the same table - it is a self-referential relationship

manager is also an Employee so it is in the Employee table

this creates a hierarchical structure




ensure fk value must exist in referenced table
establish link between tables
uniquely identify each row
no two rows have same pk
referential integrity
foreign key
uniqueness
properties of primary & foreign key
primary key
duplicates and nulls allowed
non-null
stability over time