**Introduction to Relational Databases in SQL**

# Attribute Constraints – e.g. data types on columns.

## **Query information schema tables**: information about all tables in your current database

SELECT table\_name

FROM information\_schema.tables

WHERE table\_schema = 'public';

## **Query information schema columns**: information about all columns in all of the tables in your current database

SELECT column\_name, data\_type

FROM information\_schema.columns

WHERE table\_name = 'table\_name'

AND table\_schema = 'public';

## Creating simple tables:

CREATE TABLE table\_name (

column\_a data\_type,

column\_b data\_type,

column\_c data\_type

);

## Adding column names:

ALTER TABLE table\_nameADD COLUMN column\_name data\_type;

## Insert distinct records from a table to a table:

INSERT INTO target\_table

SELECT DISTINCT column\_a, column\_b

FROM source\_table

## Insert distinct records into a table:

INSERT INTO table\_name (column\_a, column\_b)

VALUES (“value\_a”, “value\_b”)

## Rename a column:

ALTER TABLE table\_name

RENAME COLUMN old\_name, new\_name

## Drop a column:

ALTER TABLE table\_name

DROP COLUMN column\_name

## Drop Table

DROP TABLE table\_name

## Constraints

**Referential integrity Constraints – enforced through foreign keys.**

## Type Cast

CREATE TABLE weather (

temperature integer, wind\_speed text);

SELECT temperature \* CAST(wind\_speed AS integer) AS wind\_chill

## Common Data Types

text

varchar [ (x) ] – max x characters

char [ (x) ] – fixed length string of x characters

boolean – TRUE, FALSE, NULL (unknown)

date, time, timestamp

numeric [ (x, y) ] - precision of x and scale of y (total x digits and y digits after .)

integer – whole numbers

## Alter data type

ALTER TABLE table\_name

ALTER COLUMN column\_name

TYPE new\_dtype

## Truncate or transform values before altering

ALTER TABLE table\_name

ALTER COLUMN column\_name

TYPE new\_dtype

USING some\_function (column\_name)

## Example – Convert types using a function

ALTER TABLE table\_name

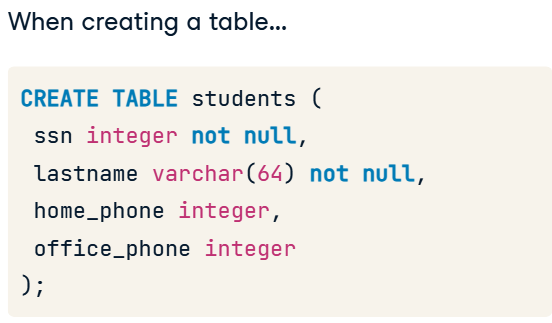
ALTER COLUMN column\_name

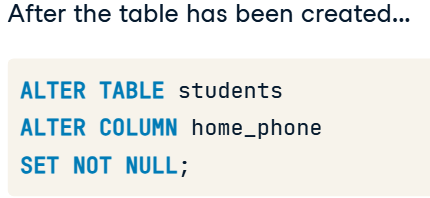
TYPE varchar(x)

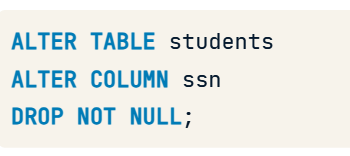
USING SUBSTRING(column\_name FROM 1 FOR x)

*Because you want to reserve only x characters for column\_name, you have to retain a SUBSTRING of every value, i.e. the first x characters of it, and throw away the rest. This way, the values will fit the varchar(x) requirement.*

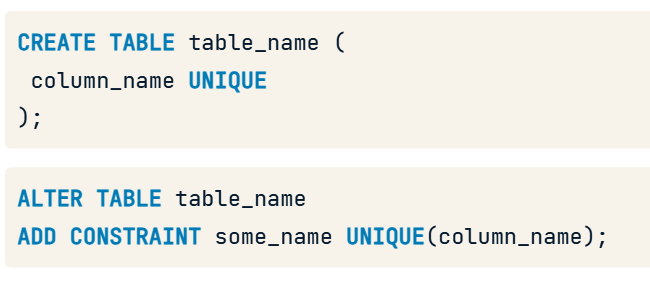
## Add or Remove Not-Null constraint







## Adding Unique constraint – name the constraint



# Key Constraints - e.g. primary keys.

In the entity-relationship diagram, keys are denoted by underlined attribute names.

## What are keys an d superkeys

The combination of all attributes is a key in itself. However, it's not called a key, but a superkey, if attributes from that combination can be removed, and the attributes still uniquely identify records.

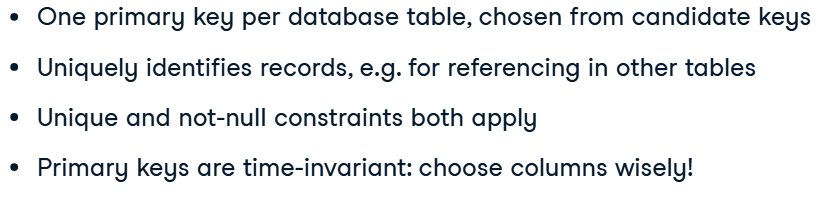
If all possible attributes have been removed but the records are still uniquely identifiable by the remaining attributes, we speak of a minimal superkey. This is the actual key. So a key is always minimal.

## Identify keys with SELECT COUNT DISTINCT

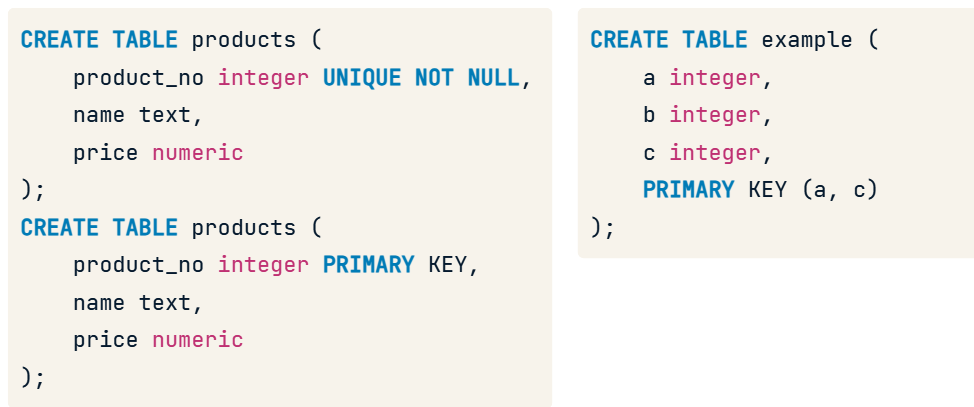
There's a very basic way of finding out what qualifies for a key in an existing, populated table:

1. Count the distinct records for all possible combinations of columns. If the resulting number x equals the number of all rows in the table for a combination, you have discovered a superkey.
2. Then remove one column after another until you can no longer remove columns without seeing the number x decrease. If that is the case, you have discovered a (candidate) key.

## Primary keys



## Specify Primary Key

);

One primary key with a,c combination