SQL Fundamentals - 3

ADVANCED SQL COMMANDS

1) Timestamps & EXTRACT
2) Math Functions
3) String Functions
4) Sub-query
5) Self Joins

ADVANCED SQL COMMANDS

TIMESTAMPS

1) TIME : Contains only Time
2) DATE : Contains only Date
3) TIMESTAMP : Contains Time and Date
4) TIMESTAMPTZ : Contains Time, Date and Time Zone

ADVANCED SQL COMMANDS

EXTRACT : Extracts the following from a date_column

1) YEAR
2) MONTH
3) DAY
4) WEEK
5) QUARTER

AGE : Calculates and returns the current age given a timestamp

TO_CHAR : Converts date types to text and is useful for formatting

ADVANCED SQL COMMANDS

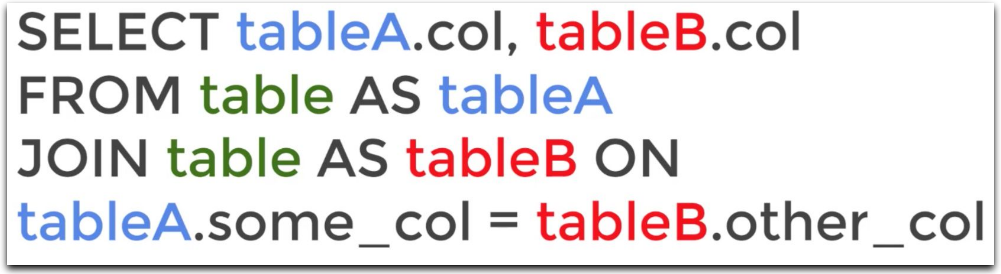
SUB-QUERY

Allows you to construct complex queries, essentially performing a query on the results of another query

SELF JOIN

Query in which a table is joined to itself. They are useful for comparing values in a column of rows within the same table.

In a self join it is compulsory to use an alias for a table.



```
SELECT tableA.col, tableB.col
FROM table AS tableA
JOIN table AS tableB ON
tableA.some_col = tableB.other_col
```

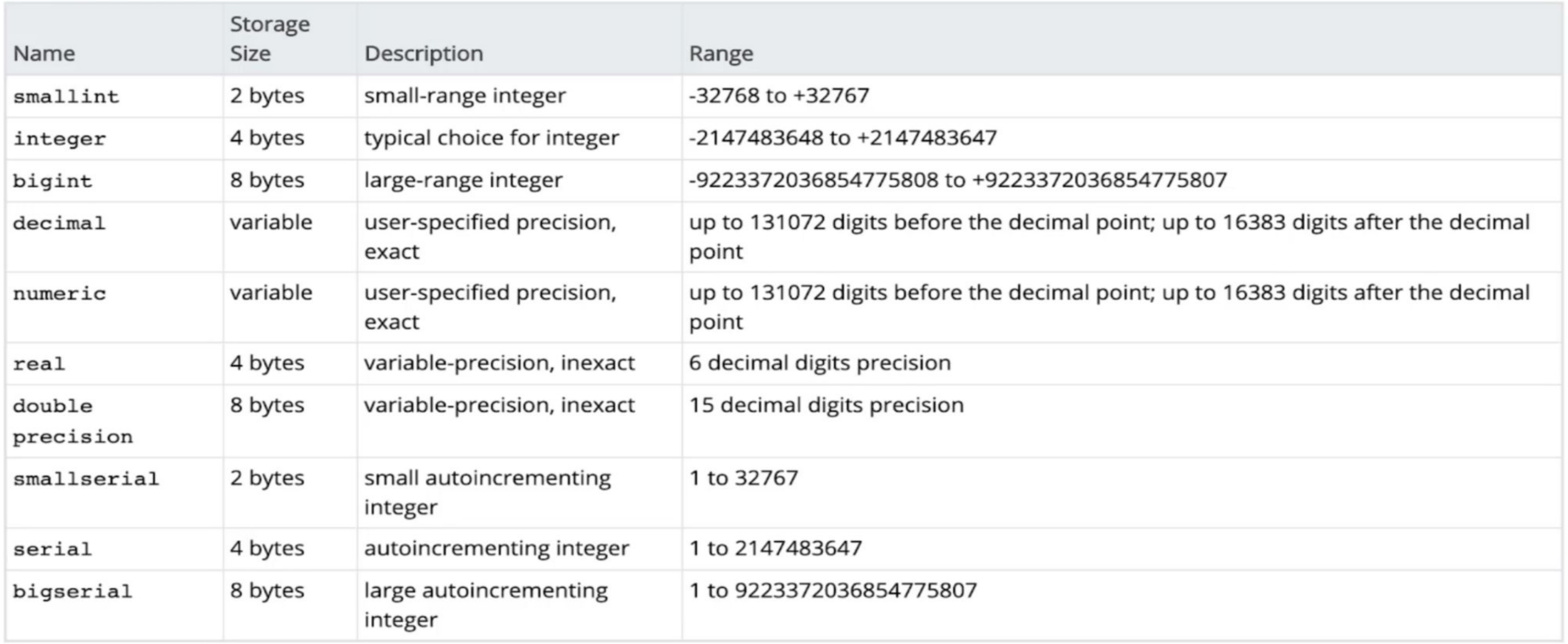
CREATING DATABASES & TABLES

DATA TYPES

1) Boolean : True or False
2) Character : Char, Varchar or Text
3) Numeric : Integral or Floating Point Numbers
4) Temporary : Date, Time, Timestamp and Interval
5) UUID : Universally Unique Identifier
6) Array : Stores an array of strings or numbers
7) JSON
8) Hstore : Key Value pair
9) Special : Geometrical Data or Network Address

CREATING DATABASES & TABLES

DATA TYPES



Name	Storage Size	Description	Range
`smallint`	2 bytes	small-range integer	-32768 to +32767
`integer`	4 bytes	typical choice for integer	-2147483648 to +2147483647
`bigint`	8 bytes	large-range integer	-9223372036854775808 to +9223372036854775807
`decimal`	variable	user-specified precision, exact	up to 131072 digits before the decimal point; up to 16383 digits after the decimal point
`numeric`	variable	user-specified precision, exact	up to 131072 digits before the decimal point; up to 16383 digits after the decimal point
`real`	4 bytes	variable-precision, inexact	6 decimal digits precision
`double precision`	8 bytes	variable-precision, inexact	15 decimal digits precision
`smallserial`	2 bytes	small autoincrementing integer	1 to 32767
`serial`	4 bytes	autoincrementing integer	1 to 2147483647
`bigserial`	8 bytes	large autoincrementing integer	1 to 9223372036854775807

CREATING DATABASES & TABLES

PRIMARY & FOREIGN KEY

1) Primary Key : Column or Group of Columns that are used to uniquely identify a row in a table. They allow us to easily discern what columns are to be used when joining tables
2) Foreign Key : Field or a Group of Fields in a table that uniquely identifies a row in another table. It is defined in a table that references to the primary key of another table.

   i) Parent Table : Table to which FK references

   ii) Child Table : Table that contains the FK

CREATING DATABASES & TABLES

CONSTRAINTS

Constraints are rules enforced on data columns in tables and are used to prevent invalid data being entered in a table. Constraints can be divided into 2 types : COLUMN CONSTRAINTS or TABLE CONSTRAINTS.

1) NOT NULL
2) UNIQUE
3) PRIMARY KEY
4) FOREIGN KEY
5) CHECK

CREATING DATABASES & TABLES

TABLE QUERIES

1) CREATE TABLE
2) INSERT
3) UPDATE
4) DELETE
5) ALTER TABLE
6) DROP TABLE
7) CHECK CONSTRAINTS

CONDITIONAL EXPRESSIONS & PROCEDURES

CASE

Used only to execute statements when certain conditions are met. Very similar to IF/ ELSE.

1) GENERAL CASE : Allows us to do all kinds of conditional checks
2) CASE EXPRESSION : Allows us to do only certain checks

CONDITIONAL EXPRESSIONS & PROCEDURES

COALESCE

Accepts unlimited number of arguments and returns the first argument which is not NULL. If all arguments are NULL, then COALESCE function will return NULL.

COALESCE(arg_1, arg_2…...arg_n)

COALESCE becomes useful when you are querying a table with NULL values and substituting the NULL values with a variable.

PRICE	DISCOUNT
100	20
200	NULL
300	30

CONDITIONAL EXPRESSIONS & PROCEDURES

COALESCE

Let’s find the FINAL PRICE

SELECT (PRICE - COALESCE(DISCOUNT,0)) AS FINAL_PRICE FROM table

PRICE	DISCOUNT	FINAL_PRICE
100	20	80
200	NULL	200
300	30	270

CONDITIONAL EXPRESSIONS & PROCEDURES

CAST

Let’s you convert one Data Type into another.

However not every instance of a data type can be cast into another data type unless it is not reasonable.

CONDITIONAL EXPRESSIONS & PROCEDURES

NULLIF

This returns a NULL Value if the arguments inside NULLIF() are equal.

NULLIF(10,10) -> NULL

NULLIF(10,12) -> 10   (Returns the first argument if arguments are not equal)

STUDENT	BRANCH
X	A
Y	A
Z	B

CONDITIONAL EXPRESSIONS & PROCEDURES

VIEWS

Often there are specific combinations of tables and conditions that you use again and again. Instead of performing the same query every time as a starting point you can create a VIEW to quickly see this query with a simple call.

View can be accessed as a virtual table and it does not store the data physically. It simply stores the query.