

SQL Visuals For SQL Sprinters

ANATOMY OF A SQL QUERY



ANATOMY OF A SQL INSERT

INSERT INTO table (col_1, col_2, col_3, ..., col_n)

VALUES (value_1, value_2, value_3, ..., value_n)

(or a query statement that returns the needed values)

ANATOMY OF A SQL UPDATE

UPDATE table (col_1, col_2, col_3, ..., col_n)

SET (col_1 = value_1, col_2 = value_2, ..., col_n = value_n)

WHERE (conditions like those in a query statement)

Be **EXTRA** careful to use a **WHERE** condition on **UPDATES**

SQL DATABASES TABLES VIEWS

CREATE DATABASE database_name;

DROP DATABASE database_name;

BACKUP DATABASE database_name
TO DISK = 'filepath';

constraints:

NOT NULL
UNIQUE
PRIMARY KEY
IDENTITY(1,1) (SQL Server Auto Increment)
int **FOREIGN KEY REFERENCES** table(ID_col)
CHECK (Age>=18)
DEFAULT value

CREATE [UNIQUE] INDEX index_name
ON table_name (column1, column2, ...);

DROP INDEX table_name.index_name;

CREATE TABLE table_name (
column1 datatype **constraint**,
column2 datatype **constraint**,
column3 datatype **constraint**,
...);

DROP TABLE table_name;

ALTER TABLE table_name
ADD
col_name datatype **constraint**;

ALTER TABLE table_name
ALTER COLUMN
col_name datatype **constraint**;

ALTER TABLE table_name
DROP COLUMN col_name;

CREATE VIEW view_name
AS (sql query statement);

SQL STORED PROCEDURES

CREATE PROCEDURE procedure_name

AS sql_statement

GO;

EXEC procedure_name;

SQL DELETE

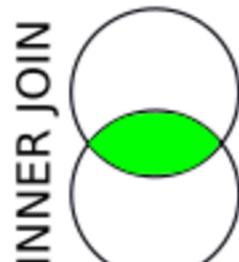
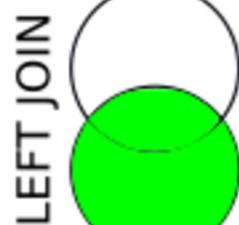
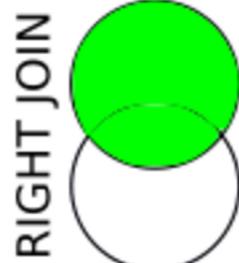
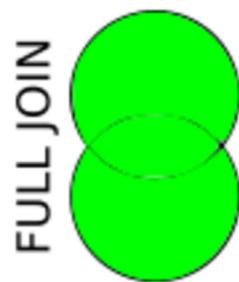
DELETE FROM table

WHERE condition;

Be **EXTRA** careful to use a **WHERE** condition on **DELETES**

ANATOMY OF A SQL QUERY

JOIN **TYPEs**



SELECT

TOP num | **TOP** pc **PERCENT** col_list or *

[**DISTINCT**] * cols col **AS** alias
MAX(col) **MIN**(col) **COUNT**(col)
AVG(col) **SUM**(col)

IFNULL(col, 0)

FROM

table some_result

table1 t1 **TYPE JOIN** table2 t2 **ON** t1.col = t2.col

TYPEs: **INNER**, **LEFT**, **RIGHT**, **FULL**, self

WHERE

condition **OR** condition condition **AND** condition
NOT condition col **LIKE** pattern col **IN** (set)
col **BETWEEN** val1 **AND** val2

val IS NULL or
val IS NOT NULL

optional

GROUP BY

col(s)

HAVING

condition

```
CASE
WHEN condition1 THEN result1
WHEN condition2 THEN result2
WHEN conditionN THEN resultN
ELSE result
END;
```

ORDER BY

col1, col2, ... ASC | DESC

ANATOMY OF A SQL INSERT

INSERT INTO table (col_1, col_2, col_3, ..., col_n)

VALUES (value_1, value_2, value_3, ..., value_n)

(or a *query statement that returns the needed values*)

ANATOMY OF A SQL UPDATE

UPDATE table (col_1, col_2, col_3, ..., col_n)

SET (col_1 = value_1, col_2 = value_2, ..., col_n = value_n)

WHERE (*conditions like those in a query statement*)

SQL DATABASES TABLES VIEWS

CREATE DATABASE database_name;

DROP DATABASE database_name;

BACKUP DATABASE database_name
TO DISK = 'filepath';

constraints:

NOT NULL

UNIQUE

PRIMARY KEY

IDENTITY(1,1) (SQL Server Auto Increment)

int FOREIGN KEY REFERENCES table(ID_col)

CHECK (Age>=18)

DEFAULT value

CREATE [UNIQUE] INDEX index_name
ON table_name (column1, column2, ...);

DROP INDEX table_name.index_name;

CREATE TABLE table_name (
column1 datatype constraint,
column2 datatype constraint,
column3 datatype constraint,
...);

DROP TABLE table_name;

ALTER TABLE table_name
ADD
col_name datatype constraint;

ALTER TABLE table_name
ALTER COLUMN
col_name datatype constraint;

ALTER TABLE table_name
DROP COLUMN col_name;

CREATE VIEW view_name
AS (sql query statement);

SQL STORED PROCEDURES

```
CREATE PROCEDURE procedure_name  
AS    sql_statement  
GO;  
EXEC   procedure_name;
```

SQL DELETE

```
DELETE FROM  table  
WHERE    condition;
```

Be **EXTRA** careful to use a **WHERE** condition on **DELETEs**