

Constraints

Keys & Foreign Keys, Referential Integrity, CHECK

Constraints

- Specify the values an attribute can take in a tuple or a table
- Specified on top of the attribute data type
- Defined in **CREATE TABLE** statement
 - https://sqlite.org/lang_createtable.html
 - https://sqlite.org/lang_conflict.html
- **Automatically checked by the database for every modification (I/U/D) operation**
 - I/U/D operations incur overhead (are slower)

Example 1

```
CREATE TABLE Product (  
    maker CHAR(32) DEFAULT('Unknown'),  
    model INTEGER PRIMARY KEY,  
    type VARCHAR(20) NOT NULL,  
    CHECK (type IN ('pc', 'laptop', 'printer')),  
    CHECK ((type = 'pc' AND model >= 1000 AND model <  
2000) OR (type = 'laptop' AND model >= 2000 AND model  
< 3000) OR (type = 'printer' AND model >= 3000 AND  
model < 4000))  
)
```

PRIMARY KEY

- Key
 - Attribute (or set of attributes) that have unique (different) values across all the tuples
 - There are no two different tuples which have the same value for the key attribute
- SQLite
 - Tuples are sorted on this attribute(s)
 - There is an index on PRIMARY KEY

NOT NULL

- The value of a NOT NULL attribute cannot be NULL
- `insert into Product(model) values(1100)`
- `insert into Product(model, maker) values(1100, 'A')`
- `insert into Product(maker, type) values('A', 'pc')`
 - *model* is primary key
 - *model* is set to `MAX(model)+1` (AUTO-INCREMENT)

DEFAULT

- When the attribute value is not specified, the DEFAULT value is used
- insert into Product(model, maker) values(1100, 'A')
 - Unknown|1100|pc

CHECK Clause

- CHECK (type IN ('pc', 'laptop', 'printer'))
- CHECK ((type = 'pc' AND model >= 1000 AND model < 2000) OR (type = 'laptop' AND model >= 2000 AND model < 3000) OR (type = 'printer' AND model >= 3000 AND model < 4000))
 - Any valid condition that can be in the *WHERE* clause
- insert into Product values('A', 1100, 'PC')
- insert into Product values('A', 2100, 'pc')

Example 2

```
CREATE TABLE PC (  
    model INTEGER REFERENCES Product(model),  
    speed FLOAT,  
    ram INTEGER,  
    hd INTEGER,  
    price DECIMAL(7,2) NOT NULL,  
    PRIMARY KEY(model)  
)
```


REFERENCES

- Foreign Key Referential Integrity
 - Cross-table attribute value constraint
 - The value of the attribute has to be one of the values of the referenced attribute (or NULL)
- SQLite
 - <https://sqlite.org/foreignkeys.html>
 - PRAGMA foreign_keys → ON/OFF
 - The referenced attribute has to be PRIMARY KEY or UNIQUE INDEX
 - model INTEGER PRIMARY KEY
 - CREATE UNIQUE INDEX product_idx_model ON Product(model)
 - For efficiency, an index should be defined on the FOREIGN KEY attribute, if not already a PRIMARY KEY
 - CREATE INDEX pc_idx_model ON PC(model)

Referential Integrity Operations

- model INTEGER REFERENCES Product(model)
 - **Product**
 - INSERT INTO Product VALUES('A', 1001, 'pc')
 - DELETE FROM Product WHERE model = 1001
 - **PC**
 - INSERT INTO PC(model, speed, ram, hd, price) VALUES(1001, 2.66, 1024, 250, 2114)
 - DELETE FROM PC WHERE model = 1001
- **UPDATE = DELETE + INSERT**

DEFERRED FOREIGN KEY

- model INTEGER REFERENCES Product(model)
 - INSERT INTO PC(model, speed, ram, hd, price)
VALUES(1001, 2.66, 1024, 250, 2114)
 - INSERT INTO Product VALUES('A', 1001, 'pc')
- model INTEGER REFERENCES Product(model)
DEFERRABLE INITIALLY DEFERRED
 - BEGIN TRANSACTION
 - INSERT INTO PC(model, speed, ram, hd, price)
VALUES(1001, 2.66, 1024, 250, 2114)
 - INSERT INTO Product VALUES('A', 1001, 'pc')
 - COMMIT TRANSACTION

ON DELETE/UPDATE Actions

- model INTEGER REFERENCES Product(model)
ON DELETE CASCADE ON UPDATE SET NULL
- Impacts operations on **Product**
- SQLite
 - NO ACTION
 - RESTRICT
 - **SET NULL**
 - SET DEFAULT
 - **CASCADE**

Constraints Summary

- Enforce data are clean & consistent
- Database is automatically guaranteeing constraints' satisfaction
 - Overhead for modification operations (I/U/D)
 - **275 ms vs 298 ms** on small Computers database
- Modification operation (I/U/D) order becomes very important
 - Error messages are hard to understand