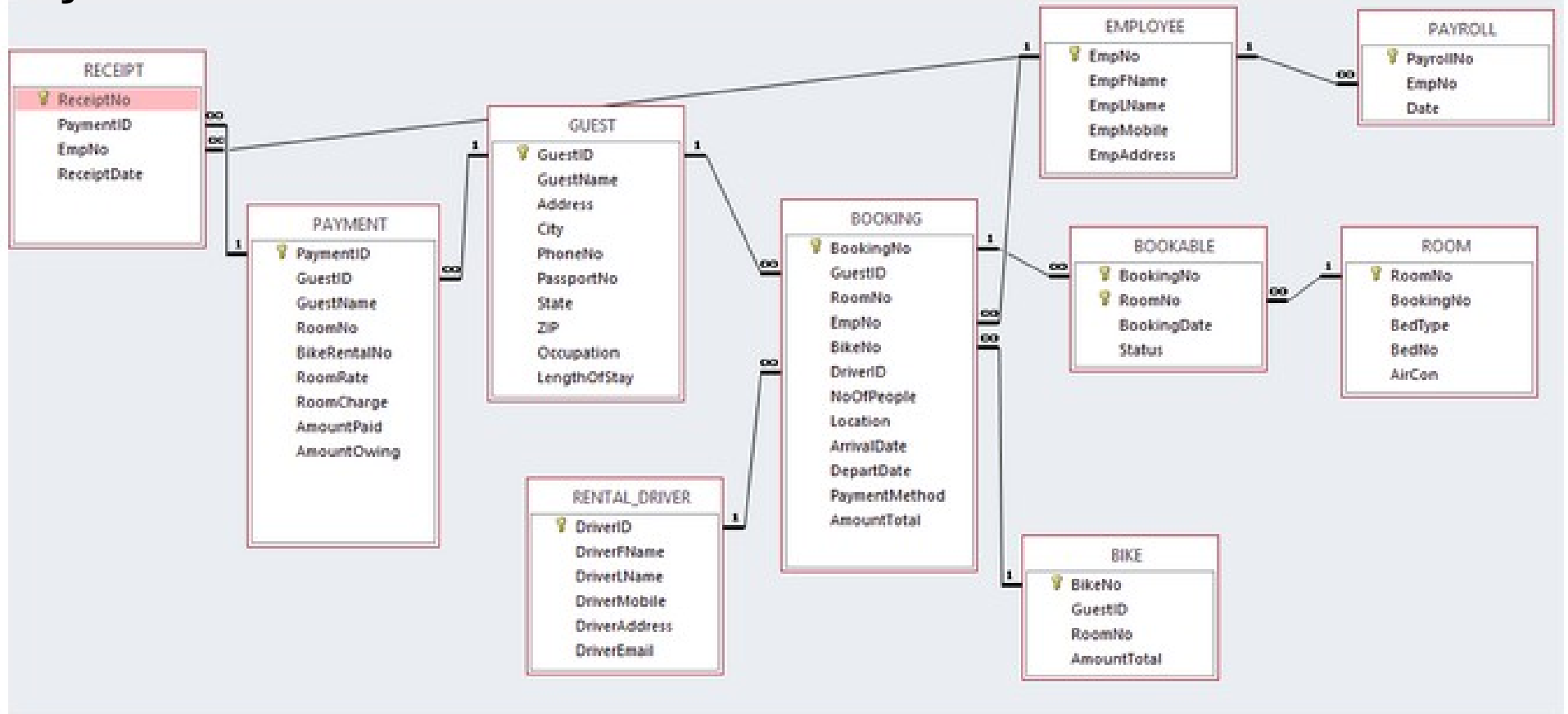


SQL veritabanları

İlişkisel veritabanı...



Birbiri ile bağlantılı tablolar



[Home](#) [About](#) [Documentation](#) [Download](#) [License](#) [Support](#) [Purchase](#)

SQLite is a [self-contained](#), [high-reliability](#), [embedded](#), [full-featured](#), [public-domain](#), SQL database engine. SQLite is the [most used](#) database engine in the world. [More Info](#)

Latest Release: [Version 3.25.2](#) (2018-09-25). [Download](#) [Prior Releases](#)

Sponsors

Ongoing development and support of SQLite is made possible in part by [SQLite Consortium](#) members, including:



```
CREATE TABLE IF NOT EXISTS records(  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  accession TEXT,  
  dscription TEXT,  
  organism TEXT,  
  seq TEXT  
);
```

```
CREATE TABLE IF NOT EXISTS features(  
  id INTEGER PRIMARY KEY,  
  featureName TEXT,  
  startPos INTEGER,  
  endPos INTEGER,  
  accession TEXT  
);
```

```
CREATE TABLE IF NOT EXISTS qualifiers(  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  featureId INTEGER,  
  qualName TEXT,  
  qualValue TEXT  
);
```

```
--@insert_records
```

```
INSERT INTO records(accession, dscription,organism, seq) VALUES (?,?,,?)
```

```
--@insert_features
```

```
INSERT INTO features(id, featureName, startPos, endPos, accession) VALUES (?,?,,?,?)
```

```
--@insert_qualifiers
```

```
INSERT INTO qualifiers(featureId, qualName, qualValue) VALUES (?,?,,?)
```

```
class SqlBlocks(object):
    """
    Loads and parses *.sql file into separate sql command blocks/snippets.
    Each block begins with standart sql comment tag followed by @
    and immediately by block name. Example:
        --@block_name
        SELECT * FROM table
        ...
        --@another_block
        ...
    The sql code are stored in standard python dictionary, where the keys are
    block names and the values are strings
    """
```

```
def __init__(self, sqlFilename):
    ''' Loads and parses SQL blocks seperated by --@@block_name tags
    :type sqlFilename: str
    '''
    self.blocks = {} # dictionary of sql code blocks
    sqlFile = open(sqlFilename, 'r')
    blockName = ''
    blockCode = ''
    line = sqlFile.readline()
    while line:
        if '--@' in line:
            ll = line.split('@', 1)
            if len(ll) > 1:
                if blockName:
                    self.blocks[blockName.strip()] = blockCode
                    blockCode = ''
                    blockName = ll[1]
                else:
                    blockName = ''
                    blockCode = ''
                    print('ERROR: No name provided for the block!')
            else:
                blockCode = blockCode + line
            line = sqlFile.readline()
        if blockName:
            self.blocks[blockName.strip()] = blockCode

    sqlFile.close()
```

```
def dump(self):
    for key in self.blocks:
        print(key)
        print(self.blocks[key])
        print('\n\n\n')

def getBlock(self, blockname):
    return self.blocks[blockname]

def __getitem__(self, key):
    return self.blocks[key]
```



ID A04321; SV 1; linear; unassigned RNA; PAT; VRL; 9193 BP.
XX
AC A04321;
XX
DT 14-JUL-1993 (Rel. 36, Created)
DT 14-APR-2005 (Rel. 83, Last updated, Version 5)
XX
DE Human immunodeficiency virus RNA for gag, env, pol, and orfO
XX
KW env protein; gag protein; pol protein.
XX
OS Human immunodeficiency virus 1
OC Viruses; Retro-transcribing viruses; Retroviridae; Orthoretrovirinae;
OC Lentivirus; Primate lentivirus group.
XX
RN [1]
RP 1-9193
RA ;
RT "ENVELOPE ANTIGENS OF LYMPHADENOPATHY ASSOCIATED VIRUS AND THEIR
RT APPLICATIONS";
RL Patent number W08602383-A1/37, 24-APR-1986.
XX
DR MD5; 6b2f2670b1e6042cfeef487908d5ce6e.
DR EuropePMC; PMC150628; 12634388.
DR EuropePMC; PMC2862539; 20104214.
DR EuropePMC; PMC2885272; 20360268.
DR EuropePMC; PMC3346622; 22290950.
DR EuropePMC; PMC5370107; 28350835.
XX
FH Key Location/Qualifiers
FH
FT source 1..9193
FT /organism="Human immunodeficiency virus 1"
FT /mol_type="unassigned RNA"
FT /db_xref="taxon:11676"
FT CDS 336..1838
FT /gene="gag"
FT /protein_id="CAA00349.1"
FT /translation="MGARASVLSGGELDRWEKIRLRPGGKKKYKLVKLVASRELERFA
FT VNPGLLETSEGCRIQLGQLQPSLQGTGSEELRSLYNTVATLYCVHQRIEIKDTKEALDKI
FT EEEQNKSKKKAQQAAADTGHSSQVSQNYPIVQNIQGMVHQAI SPRTLNAWVKVVEEKA

File Edit View Help

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

Create Table Create Index Modify Table »

Name	Type	Schema
▼ Tables (4)		
▶ features	CREATE TABLE	
▶ qualifiers	CREATE TABLE	
▶ records	CREATE TABLE	
▶ sqlite_sequence	CREATE TABLE	
Indices (0)		
Views (0)		
Triggers (0)		

Edit Database Cell

Mode: Text

Import

Export

Set as NULL

Empty text area for editing the database cell content.

Type of data currently in cell: NULL
0 byte(s)

Apply

DB Schema

Name	Type	Schema
▼ Tables (4)		
▶ features	CREATE TABLE	
▶ qualifiers	CREATE TABLE	
▶ records	CREATE TABLE	
▶ sqlite_sequence	CREATE TABLE	
Indices (0)		
Views (0)		
Triggers (0)		

SQL Log

Plot

DB Schema

Remote

Table: records

	id	accession	dscription	organism	seq
1	1	A04321.1	Human imm...	Human imm...	GGTCTCTC...

Table: features

	id	featureName	startPos	endPos	accession
1	0	source	0	9193	A04321.1
2	1	CDS	33		
3	2	CDS	19		
4	3	CDS	45		
5	4	CDS	57		
6	5	CDS	83		
7	6	source	0		
8	7	repeat_region	0		
9	8	repeat_region	0		

Table: qualifiers

	id	featureId	qualName	qualValue
1	1	0	organism	Human imm...
2	2	0	mol_type	unassigned ...
3	3	0	db_xref	taxon:11676
4	4	1	gene	gag
5	5	1	protein_id	CAA00349.1
6	6	1	translation	MGARASVL...
7	7	2	gene	pol
8	8	2	protein_id	CAA00350.1

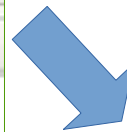


Table: features

Table: qualifiers