



A11.2 Relational databases and keys

Databases and big data · A level · OCR H446 1.3.2, AQA 7517 4.10.2,
Eduqas A500QS 2.4 · about 55 min

Name _____

Class _____

Date _____

What this lesson is about

Flat files against relational databases, and every kind of key: primary, composite, foreign and secondary, with indexes and how records are found.

- Flat files and relational databases
- The vocabulary
- Keys
- Indexes
- How records are found in a file

Questions

6 marks in all

- What is a composite primary key? [1 mark]
 - ☐ A A primary key made of two or more attributes that together are unique
 - ☐ B A primary key that is also a foreign key
 - ☐ C A key that is indexed for searching
 - ☐ D A key made by joining two tables
- Which describes a secondary key? [1 mark]
 - ☐ A An attribute other than the primary key that is indexed so records can be searched or sorted by it quickly
 - ☐ B The second attribute of a composite key
 - ☐ C A backup copy of the primary key
 - ☐ D A foreign key in a link table
- Which are disadvantages of a flat file database compared with a relational database? [1 mark]

Tick every answer that is true.

 - ☐ A The same facts are stored many times (data redundancy)
 - ☐ B A missed edit can leave the data inconsistent
 - ☐ C It needs a DBMS to be set up
 - ☐ D It cannot be opened by a spreadsheet
- Adding an index on the task field of a runs table will usually... [1 mark]
 - ☐ A make searches by task faster, but take more storage and slow down inserts, updates and deletes
 - ☐ B make every operation on the table faster
 - ☐ C stop two records having the same task
 - ☐ D remove the need for a primary key

5. Records are stored at an address calculated from their key. Which file organisation is this?

[1 mark]

- ☐ A Direct (hashed) access
- ☐ B Serial
- ☐ C Sequential
- ☐ D Indexed sequential

6. What does this print?

[1 mark]

```
SLOTS = 7
table = [None] * SLOTS
for key in [9, 16, 4]:
    address = key % SLOTS
    while table[address] is not None:
        address = (address + 1) % SLOTS
    table[address] = key
print(table)
```

The task: a composite key

A robot may make several attempts at a task, numbered from 1 for each robot. The list `attempts` holds records `(robot_id, attempt, cm)`: two whole numbers and a float. 1. Create a table `attempts` with the fields `robot_id`, `attempt` and `cm`, and the composite primary key `(robot_id, attempt)`. 2. Insert every record, in order. For each one print `added robot <robot_id> attempt <attempt>`, or, if the database refuses it with `sqlite3.IntegrityError`, print `rejected robot <robot_id> attempt <attempt>: already logged`. 3. Use `SELECT COUNT(*)` to print the number of records now in the table as `records:`
<n>. Seven lines in all. The robot does not move.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

import sqlite3

attempts = [(1, 1, 42.0), (1, 2, 45.5), (2, 1, 38.0), (1, 2, 44.0), (3, 1, 76.5), (2, 1,
39.0)]
db = sqlite3.connect(":memory:")
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/a11-2-relational-databases-and-keys/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. The Booking table has the key (StudentID, RobotID, SessionDate). Could a student book two robots on the same day? Could they book the same robot twice on one day?
2. Store 1,000 random run records, then time `SELECT` by task with and without an index. How big must the table be before the index shows?
3. A car park system looks up a car by its registration. Which file organisation would you choose, and why?