



F7.4 SQL: two tables and changing data

Files and databases · GCSE · OCR J277 2.2.3, AQA 8525 3.7.2 · about 20 min

Name _____

Class _____

Date _____

What this lesson is about

Queries across two tables; INSERT, UPDATE and DELETE, and the robot logging its own runs.

- Questions across two tables
- INSERT: adding a record
- Values from the program
- UPDATE and DELETE

Questions

5 marks in all

- In `SELECT robots.name, runs.cm FROM runs, robots WHERE runs.robot_id = robots.robot_id`, what does the WHERE condition do? [1 mark]
 - ☐ A Matches each run to the robot that did it
 - ☐ B Deletes robots with no runs
 - ☐ C Sorts the runs
 - ☐ D Counts the robots
- Which statement adds a new record? [1 mark]
 - ☐ A INSERT INTO
 - ☐ B UPDATE
 - ☐ C SELECT
 - ☐ D DELETE FROM
- What does `DELETE FROM runs` do with no WHERE? [1 mark]
 - ☐ A Deletes every record in runs
 - ☐ B Deletes nothing
 - ☐ C Deletes the table's last record
 - ☐ D Gives an error
- Complete the statement to change Ada's colour to purple: `_____ robots SET colour = 'purple'` [1 mark]
`WHERE name = 'Ada'`

- Why pass values to a query with `?` placeholders instead of joining text into the SQL? [1 mark]
 - ☐ A It stops SQL injection, where typed text changes what the query does
 - ☐ B It makes the query shorter
 - ☐ C SQL cannot contain numbers
 - ☐ D Placeholders are faster to type

The task: log and correct

Using `logbook.open_db()` : drive forward 35 cm and **insert** it as run 7 for robot 2 (Bolt), task `wall` , with the distance the robot actually covered (`position()` , rounded to 1 decimal place) and its time. **Update** run 3's task to `ramp` . **Delete** run 5. Then, with one query across both tables, print every run by Bolt, in run order, as `run 3 ramp 38.0` .

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

import logbook
db = logbook.open_db()
```

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



Do it on the robot

www.bugbotlab.com/learn/f7-4-sql-two-tables-and-changing-data/

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

Challenges

1. Find the total distance each robot has driven, by writing a query for each robot inside a Python loop over the robots table.
2. Delete every run of a robot, then the robot itself. Why does the order matter?
3. Ask the user for a colour and show that robot's runs, using a `?` placeholder.