MODULE N DATABASES AND SQL · N1

Databases: tables, records and fields: mark scheme

21 marks · OCR J277 spec 2.2.3 (the use of records to store data; use of 2D arrays to emulate database tables of a collection of fields and records), with 2.2.1 (iteration) for 3(d)

How to use. 1 mark per bullet up to the Max shown. **/** separates alternative answers. Words in (brackets) are not needed for the mark. Code answers are marked for logic, not exact syntax: accept OCR Exam Reference Language or any high-level language, and allow any logically valid method. 2D indexes may be written `[r,c]` or `[r][c]`, but must be in the row, column order given in the question. Minor syntax slips are not penalised; a repeated slip is penalised once then follow through. Do not accept flowcharts, structured English or bullet points for 3(d). Credit any other valid answer that shows the same understanding.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) J277 2.2.3	All of the data about one item, one row: Record One category of data stored for every item, one column: Field A collection of records with the same fields: Table	**3**	1 mark per correct row. More than one box ticked in a row: 0 for that row.
2(a) J277 2.2.3	7	**1**	One record per bike (B101 to B107). Do not accept 8 (the heading row is not a record) or 6.
2(b) J277 2.2.3	6	**1**	BikeID, Type, Colour, Size, DailyRate, Electric. Do not accept 7.
2(c) J277 2.2.3	B102, B105, B107	**2**	2 marks for all three and nothing else. 1 mark for any two correct and nothing else, or all three correct plus one extra. Any order. Common wrong extras: B106 (Size M) and B104 (not electric).
2(d) J277 2.2.3	1 mark each to max 2: A new field / column is added (to the table) … … with a suitable name, e.g. `LastService` / `ServiceDate` Every record / each bike / all 7 records will store a (date) value in the new field The number of fields goes up to 7 / the number of records stays at 7	**2**	The first mark needs the idea of a new field or column. "Add a new record / row": 0 for the whole part (a record is one bike, not one item of data). Do not accept a name with spaces for the name mark (e.g. "Last Service"); accept underscores.

QUESTION	ANSWER	MARKS	GUIDANCE
2(e) J277 2.2.3	1 mark each: `BikeID`, `Type` and `Colour` all given with suitable values (e.g. B108, Road, Grey) `Size` given as S / M / L, `DailyRate` as a price / number (e.g. 20.00), `Electric` as True / False Example: B108 | Road | Grey | S | 20.00 | False	**2**	Every cell must hold a value: a missing value loses the mark for that group. BikeID: accept any ID in the same form (B followed by digits); do not penalise an ID already in the table (primary keys are not on J277). Accept a type not in the table (e.g. Tandem, Folding). Accept DailyRate with or without a £ sign or decimal places. Accept Yes / No for Electric.
3(a) J277 2.2.3	1 mark each: Row: one record / all the data about one bike Column: one field / one item of data (e.g. the colour) for every bike	**2**	Accept answers using the bike data, e.g. "row 0 is bike B101" / "column 2 is the Colour field". Reversed answers (row = field, column = record): 0.
3(b) J277 2.2.3	`Road`	**1**	Row 4 is bike B105; column 1 is Type. Accept with or without quotation marks. Do not accept "B104" or "Mountain" (counting from 1), or "Black".
3(c) J277 2.2.3	`bikes[2,2] = "Yellow"`	**1**	Accept `bikes[2][2] = "Yellow"`. Indexes must be 2 then 2 (B103 is row 2; Colour is column 2). Allow missing quotes and any capitals for Yellow. Do not accept ==, or the value on the left (`"Yellow" = bikes[2,2]`).

QUESTION	ANSWER	MARKS	GUIDANCE

3(d)
J277 2.2.3
2.2.1
	1 mark each to max 3:
	Loop over every record / row 0 to 6
	Checks the Electric field of the current row: `bikes[r,5] == "True"`
	Outputs the BikeID field of the same row: `print(bikes[r,0])` (inside the IF)
	Example:
	`for r = 0 to 6`
	`  if bikes[r,5] == "True" then`
	`  print(bikes[r,0])`
	`  endif`
	`next r`
		3	Expected output (not required): B102, B105, B106, B107. The check and output marks need the loop variable as the first (row) index. Accept a while loop with a counter, and `bikes.length - 1` / `len(bikes)` as the loop limit. Loop 1 to 7 or 0 to 7: lose the loop mark, FT for the other two. Comparing with Boolean `True` instead of the string `"True"`: allow. Checking rows one at a time with no loop: max 1 (iteration is required). Outputting the whole record instead of the BikeID: lose the output mark.

4(a) J277 2.2.3	450	**1**	One record per variety. Do not accept 8 or 3600.
4(b) J277 2.2.3	1 mark each: Records: 480 Fields: 9	**2**	450 + 30 = 480 records; 8 + 1 = 9 fields. Adding records does not change the number of fields. Answers in the wrong boxes (Records 9, Fields 480): 0.

Total: 21 marks