CAMBRIDGE IGCSE (0478)

Module N: Databases and SQL

Workbooklet mark scheme · 5 parts · 103 marks

Contents

Part	Topic	Marks
N1	Databases: tables, records and fields: mark scheme	20
N2	Database data types and validation: mark scheme	22
N3	Primary keys: mark scheme	20
N4	SQL: SELECT, FROM and WHERE: mark scheme	20
N5	SQL: ORDER BY, SUM, COUNT, AND and OR: mark scheme	21
	Total	**103**

MODULE N DATABASES AND SQL · N1

Databases: tables, records and fields: mark scheme

20 marks · Cambridge 0478 syllabus 9.1 (define a single-table database from given data storage requirements, including fields and records)

How to use. One mark per creditworthy point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Ignore minor slips in spelling and capitals unless the guidance says otherwise. Credit any other valid answer that shows the same understanding.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 9.1	One mark for each correct row: A field is one row … about one item: False Every record in a table has the same fields: True Adding a new record also adds a new field: False	**3**	Row 1 describes a record. Row 3: a new record is a new row; the fields stay the same. Both boxes ticked in a row: 0 for that row.
2(a) 0478 9.1	6	**1**	Do not accept 8 (the number of records).
2(b) 0478 9.1	8	**1**	T01 to T08. Do not accept 9 (the heading row is not a record) or 6.
2(c) 0478 9.1	`Category`	**1**	Field name must be given as it appears; ignore capitals. Do not accept `ToolName`, or a data value such as "Garden".
2(d) 0478 9.1	T02, T06, T07	**2**	Two marks for all three and nothing else. One mark for any two correct and nothing else, or all three correct plus one extra. Any order. T07 has a deposit of exactly 50.00, so it is included ("50.00 or more"). Common wrong extras: T03, T04 (available today).
3(a) 0478 9.1	Six correct field names: 3 marks // four or five: 2 marks // two or three: 1 mark Item 1: `MemberNumber` // `MembershipNo` // `MemberID` Item 2: `FirstName` // `Forename` Item 3: `FamilyName` // `Surname` // `LastName` Item 4: `Instrument` Item 5: `MobileNumber` // `PhoneNumber` // `Mobile` Item 6: `FeePaid` // `PaidFee` // `Paid`	**3**	Accept any meaningful name for the item, written with no spaces (accept underscores, e.g. `First_Name`). A name containing a space is not correct. Do not accept a data value as a name (e.g. "Violin") or a name that does not match the item (e.g. `Name` for item 3 alone, `Date` for item 6). The same name for two items: credit it once only.

QUESTION	ANSWER	MARKS	GUIDANCE
3(b) 0478 9.1	One mark for each complete, suitable record: Record 1: a value in all six columns, each suitable for its item Record 2: a value in all six columns, each suitable for its item, and a different membership number from record 1 Example: M001 | Rosa | Okonkwo | Cello | 07700 900123 | TRUE M002 | Idris | Varga | Flute | 07700 900456 | FALSE	**2**	Item 1 for record 2 must differ from record 1 (the stem says it is different for every musician). Accept any made-up names, instruments and phone numbers in any sensible format. Item 6: accept TRUE / FALSE // Yes / No // Y / N. A value in the wrong column (e.g. an instrument in item 3): lose the mark for that record. Any blank cell: lose the mark for that record.
3(c) 0478 9.1	Add a new field (to store the date joined) // add a field such as `DateJoined`	**1**	Accept "add a new column for the date joined". Do not accept "add a new record / row", or "add the date to the first name field".
3(d) 0478 9.1	One mark for each: Fields: 7 Records: 64	**2**	6 items + the new field from (c) = 7 fields; one record per musician. Fields 6 (change in (c) ignored): 0 for that mark. Answers on the wrong lines (Fields 64, Records 7): 0.
4(a) 0478 9.1	Record, max 2: All the data stored about one item / one tool // one row (of the table) Example: any one whole row of `TOOL`, e.g. T04, Tile cutter, Building, 18.00, 60.00, TRUE // "the row for the tile cutter" Field, max 2: One item / type of data stored for every item / tool // one column (of the table) // a named category of data, e.g. a price Example: any one field name from `TOOL`, e.g. `DailyCharge` // a whole column, e.g. "the Category column"	**4**	The example mark can be awarded even if the description mark is not. A single value (e.g. "Tile cutter" or "18.00") is not an example of a record or a field. Reversed descriptions (record = column, field = row): 0 for those description marks. Accept a field example given as a field name with its values, e.g. "Deposit: 40.00, 100.00, ...".

Total: 20 marks

MODULE N DATABASES AND SQL · N2

Database data types and validation: mark scheme

22 marks · Cambridge 0478 syllabus (exams 2026-2028) 9.1 (define a single-table database from given data storage requirements: fields, records, validation), 9.2 (suggest suitable basic data types: text/alphanumeric, character, Boolean, integer, real, date/time), with 7.5(a) (the need for validation checks; range, length, type, presence, format, check digit)

How to use. One mark per bullet point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Credit any other valid answer that shows the same understanding. Use the database data type names from syllabus 9.2: **text** or **alphanumeric** are both accepted for text/alphanumeric; **char** is accepted for character; **date** is accepted for date/time. Do not accept STRING, NUMBER, FLOAT, DECIMAL or CURRENCY as database data types (Cambridge examiners do not credit programming or software-specific type names for database fields).

QUESTION	ANSWER	MARKS	GUIDANCE
1 0478 9.2	One mark for each correct row: PotSize: Character Price: Real Stock: Integer Evergreen: Boolean DateArrived: Date/time	**5**	A row with more than one tick scores 0. PotSize: also accept Text/alphanumeric (one letter can be stored as text). Price: do not accept Integer (12.50 and 4.99 have a decimal part). Stock: do not accept Real (it is a count of pots, always a whole number). Evergreen: also accept Text/alphanumeric (Yes/No can be stored as text), as Cambridge mark schemes do for two-value fields; Boolean is the expected answer. DateArrived: Date/time only.
2 0478 9.2	Text/alphanumeric ... (because) the leading zero would be lost if it were stored as an integer // it contains a space // no calculations are done on a phone number	**2**	Reason mark only if Text/alphanumeric (or text / alphanumeric) is given. Do not accept Integer, Real or Character for the data type. Do not accept STRING. Do not accept "it is not a number" or "it has numbers and letters" alone (the phone number has no letters). Credit any other valid reason linked to the phone number.
3(a) 0478 7.5(a) 0478 9.1	Any one from: to check that the data entered is reasonable / sensible / acceptable to check that the data entered meets a set of rules / criteria (before it is accepted / stored) to reject data that is unreasonable / does not meet the rules (as it is entered)	**1**	Do not accept "to check the data is correct / accurate" (data can pass validation and still be wrong). Do not accept "to check that the data has been copied / typed correctly from the original" (that is verification). Do not accept "to stop errors" alone (too vague).

QUESTION	ANSWER	MARKS	GUIDANCE
3(b) 0478 7.5(a) 0478 9.1	One mark for each correct row: Stock: range check DateArrived: presence check PlantID: format check CommonName: length check Price: type check Barcode: check digit	**6**	Mark each row on its own. If a check is used more than once, credit it only in the row where it is correct. Price: also accept "character check" (seen in Cambridge mark schemes for the same idea). Barcode: accept "check digit check".
4 0478 7.5(a)	Any two from: the check digit is calculated / recalculated from the other digits (of the barcode number) when it is entered ... using the same algorithm / calculation that was used to create the check digit the calculated digit is compared with the check digit entered / the last digit if they do not match, the barcode number is rejected / an error is reported / it must be entered again if they match, the barcode number is accepted	**2**	Both marks can come from any two points. Accept a named method, e.g. modulo 11, for the algorithm point. Do not accept "the check digit checks the number is correct" alone (too vague). Do not credit a description of a parity bit or checksum sent with a transmission.
5 0478 9.1 0478 9.2 0478 7.5(a)	One mark for each correct data type: BookingRef: Text/alphanumeric Seats: Integer WorkshopDate: Date/time One mark for each suitable validation check **and** what it would test for that field: BookingRef: format check: the first character is B and the next three are digits // length check: exactly four characters // presence check: a reference must be entered / must not be empty Seats: range check: from 1 to 12 (inclusive) // type check: only a whole number / integer is accepted // presence check: a number of seats must be entered WorkshopDate: type check: must be a (valid) date // format check: e.g. DD/MM/YYYY // range check: e.g. not before today's date / within the dates the workshops run // presence check: a date must be entered	**6**	Data type and check are marked separately: a wrong data type does not stop the check mark. The check mark needs the check named **and** a test that fits that field. A check name alone, or a test alone with no check named, scores 0. Seats range: the limits must be 1 and 12; do not accept 0 to 12, "less than 12" or a range with no limits. BookingRef format: must describe the pattern (B then three digits); "must be in the right format" alone is too vague. BookingRef: do not accept Character or Integer for the data type. Seats: do not accept Real. Credit any other valid check with a rule that fits the requirement.

Total: 22 marks

MODULE N DATABASES AND SQL · N3

Primary keys: mark scheme

20 marks · Cambridge 0478 syllabus (exams 2026-2028) 9.3 (understand the purpose of a primary key and identify a suitable primary key for a given database table), with 9.1 (define a single-table database from given data storage requirements)

How to use. One mark per creditworthy point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Credit any other valid answer that shows the same understanding. Field names do not need exact capitals or spacing, but must clearly name the field.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 9.3	One mark for each correct row: A primary key uniquely identifies each record: True Two records can have the same primary key value if other fields differ: False A primary key field can only store whole numbers: False If no existing field is unique, a new field can be added to be the primary key: True	**4**	A row with both boxes ticked scores 0.
2(a) 0478 9.3	Any two from: A primary key must have a different / unique value for every record Every field (in the table) has at least one repeated value // no field has a unique value for every record A correct example from the table, e.g. two paintings are called Untitled // E. Marsh / R. Tanaka appears twice // 1968 appears twice // Oil / Acrylic appears twice // Room 3 / 5 appears twice New paintings could repeat any of these values (e.g. another painting by the same artist)	**2**	max 2. Credit an example only if the value really is repeated in the table. Only one example can be credited.
2(b) 0478 9.3	One mark for each: Field name: a suitable identifier name, e.g. PaintingID // PaintingNumber // CatalogueNumber // PaintingCode Example value: a value that could be a unique code / number for one painting, e.g. P001 // 1 // HD0001	**2**	Do not credit the name of an existing field (e.g. Title, Artist) or a name that does not suggest an identifier (e.g. Painting). Do not credit as the example value the painting's existing descriptive data (e.g. Harbour at Dawn, E. Marsh, Oil). Credit any number or code that could identify this one painting (e.g. 1, 3, P001). The example mark can be awarded even if the field name is not credited.
3(a) 0478 9.3	RunnerNumber	**1**	Do not credit FinishTime (see 3(d)) or any other field. Do not credit a value (e.g. 1032) instead of the field name.

QUESTION	ANSWER	MARKS	GUIDANCE
3(b) 0478 9.3	Any two from: To uniquely identify each record // each runner Each record / runner has a different value // no two records can have the same value ... so that one record / one runner's details can be found / selected // the correct record can be updated (e.g. when entering one runner's finish time) without confusing it with another	**2**	max 2. "Identifies a record" without the idea of unique / only one record = no mark for MP1. Do not credit "it is the first field" or "it sorts the table".
3(c) 0478 9.3	Any two from: Surname values are not unique // can be repeated // two runners can have the same surname Use of data: two runners / records have the surname Hughes // Daniel Hughes and Tom Hughes // records 1047 and 1103 ... so a surname would not identify one record // a search on Hughes would return more than one record	**2**	max 2. MP2 must quote the repeated surname (Hughes) from the table. Do not credit "names can be spelt wrongly" or "names are too long" alone.
3(d) 0478 9.3	Any two from: Two runners could finish with the same time // finish times could repeat when more records are added A primary key must be unique for every record (that will ever be stored), not only the records shown ... so a finish time might not identify one record / runner A runner has no finish time until they finish // until the race is over, so the record would have no key value	**2**	max 2. Do not credit "it is a time" or "it is not a number" alone: a primary key does not have to be a number.
3(e) 0478 9.3	Any two from: 1058 is already used // already belongs to (the record for) Priya Shah Primary key values must be unique // there cannot be a duplicate primary key value ... two records would have the same key, so 1058 would not identify one runner // a search / update using 1058 could find / change the wrong runner	**2**	max 2. Do not credit "the number is wrong" or "it is invalid" alone.
4(a) 0478 9.3, 9.1	PermitNumber	**1**	Do not credit VehicleReg (see 4(b)) or any other field.
4(b) 0478 9.3, 9.1	Any two from: Each record is a permit, not a vehicle // a vehicle can have more than one permit (when the permit is renewed) Old permit records are kept, so the same vehicle can have more than one record (in the table) ... so the same VehicleReg value would appear in more than one record // VehicleReg values would not be unique (for each record) ... so VehicleReg would not identify one permit / record	**2**	max 2. Do not credit "registrations can change" or "registrations contain letters" alone. MP3 and MP4 can be credited without MP1 or MP2.

Total: 20 marks

MODULE N DATABASES AND SQL · N4

SQL: SELECT, FROM and WHERE: mark scheme

20 marks · Cambridge 0478 syllabus 9.4 (read, understand and complete SQL scripts to query data stored in a single database table: SELECT, FROM, WHERE; identifying the output given by an SQL statement). ORDER BY, SUM, COUNT, AND and OR are covered in N5.

How to use. One mark per creditworthy point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Ignore capitals in SQL keywords and minor slips in spacing; a statement may be written on one line. Field and table names must be spelt exactly as in the table `TREES`, with no added spaces. Text and character values may be in single or double quotes (`'B'` // `"B"`); numbers must not be in quotes. Ignore a missing final semicolon. In output answers, copy the values exactly, in the column order given in the SELECT line; ignore field headings and the order of the rows. Credit any other valid answer that shows the same understanding.

QUESTION	ANSWER	MARKS	GUIDANCE
1 0478 9.4	One mark for each: • (line 1) displays only the `Variety` and `Section` fields // shows only the variety and section of each tree • (line 2) takes the data from / searches the table `TREES` • (line 3) only includes the records / trees where `Planted` is greater than 2018 // trees planted after 2018 Credit any other valid answer.	**3**	Line 1: do not accept "selects the records" or "selects the trees" (records are chosen by WHERE). Line 3: do not accept "planted in or after 2018" // `>= 2018`. (Output, not required: Pear B, Apple B, Plum B; checked in sqlite3.)
2(a) 0478 9.4	One mark for each correct row: • `T01 Apple` • `T03 Plum` • `T06 Pear`	**3**	Trees taller than 4.0 m: 4.2, 5.1 and 4.8. Each row needs the TreeID **and** the variety, in that order. Any extra row (e.g. `T05 Cherry`) or extra field (e.g. the height): max 2.
2(b) 0478 9.4	3 // three	**1**	T03, T06 and T08 are in section C. Do not accept 15 (number of values) or 5 (number of fields).
3 0478 9.4	One mark for each correct line: • `SELECT TreeID, Planted` • `FROM TREES` • `WHERE Section = 'B';`	**3**	First line: both fields, separated by a comma, in either order; do not accept `SELECT *` or extra fields (the question says "only"), or `AND` between the fields. Third line: `B` must be in quotes.

QUESTION	ANSWER	MARKS	GUIDANCE
4 0478 9.4	One mark for each correct line: • `SELECT Variety, HeightM` • `FROM TREES` • `WHERE Planted < 2016;`	**3**	Fields in either order. Accept `WHERE Planted <= 2015` (years are whole numbers) and `WHERE 2016 > Planted`. Do not accept `<= 2016`, `'2016'` in quotes, or words in place of <. Lines in the wrong order (e.g. WHERE before FROM): max 2. (Output, not required: Apple 4.2, Plum 5.1, Pear 4.8.)
5(a) 0478 9.4	One mark for each error identified **and** corrected: • Line 01: `SELECT TreeID, HeightM` • Line 02: `FROM TREES` • Line 03: `WHERE Variety = 'Plum';`	**3**	The line number alone, or a correction without its line number: 0 for that error. Line 01: a comma is needed between the fields; do not accept `AND` or `SELECT *`. Line 02: the table name is `TREES`. Line 03: `Plum` in single or double quotes.
5(b) 0478 9.4	One mark for each correct row: • `T03 5.1` • `T07 2.4`	**2**	Each row needs the TreeID and the height, in that order. Any extra row or extra field (e.g. `Plum`): max 1.

6
0478 9.4
	One mark for each:
	• `SELECT * … FROM TREES`
	• `WHERE TreeID = 'T05';`
	Example:
	`SELECT *`
	`FROM TREES`
	`WHERE TreeID = 'T05';`
		2	Accept all five fields listed, separated by commas, in place of *. `T05` must be in quotes. Do not accept `SELECT ALL` or `SELECT TreeID` alone for the first mark.

Total: 20 marks

MODULE N DATABASES AND SQL · N5

SQL: ORDER BY, SUM, COUNT, AND and OR: mark scheme

21 marks · Cambridge 0478 syllabus 9.4 (read, understand and complete SQL scripts to query a single table: SELECT, FROM, WHERE, ORDER BY ascending/descending, SUM, COUNT, AND, OR; identify the output of an SQL statement)

How to use. One mark per creditworthy point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Ignore the case of SQL keywords, a missing final semicolon and extra spaces between keywords. Field and table names must be spelt as in the table (`RepairJob`, `Category`, `Minutes`, `PartsCost`, `JobID`). Accept single or double quotation marks around text values; text values in a condition must be in quotation marks. For outputs, ignore column headings and extra spacing; do not accept quotation marks or commas added to the values. Credit any other valid answer that shows the same understanding.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 9.4	One mark for each correct row: • adds together the values: `SUM` • number of records that match: `COUNT` • largest value first: `ORDER BY` … `DESC` // `ORDER BY` … `DESCENDING` • at least one of two conditions: `OR`	**4**	Row 3 needs both `ORDER BY` and `DESC`/`DESCENDING`; `DESC` alone or `ORDER BY` alone = 0; `ASC` = 0. Accept a field name written between the keywords (e.g. `ORDER BY Minutes DESC`). Accept `SUM()` / `COUNT()` with brackets. Row 4: `AND` = 0.
2(a) 0478 9.4	One mark per bullet point: • the three items Radio, Kettle and Desk lamp only (in any order) • the correct Minutes value beside each item (60, 30, 25), and no other fields output • in the order Radio, Kettle, Desk lamp Correct output: Radio 60 Kettle 30 Desk lamp 25	**3**	Including Toaster (not fixed) or any non-Electrical job loses the first mark; the order mark can still be given if the three correct items are in the correct order relative to each other. Ascending order (Desk lamp, Kettle, Radio) loses the third mark only. Extra fields (e.g. `JobID`, `Category`) lose the second mark. Item and minutes may be on the same line or in two columns.
2(b) 0478 9.4	Any **two** from: • the `WHERE` clause / condition decides which records are output // `ORDER BY` does not filter / select records • the condition is not changed, so the same (three) repair jobs are output // Radio, Kettle and Desk lamp are still output • `ASC` only changes the order // the jobs are sorted in ascending order of Minutes instead • the shortest job / Desk lamp (25) would now be first and the longest / Radio (60) last // the order is reversed	**2**	"It only changes the order" alone = 1. Do not credit "ASC means all records are output" or answers saying a different number of records is output.

QUESTION	ANSWER	MARKS	GUIDANCE
3(a) 0478 9.4	4	**1**	Correct answer only. Records R03, R05, R07, R08 (R05 meets both conditions but is counted once). 5 (Winter coat, 40, is not > 40; or R05 counted twice) = 0. 1 (AND used) = 0.
3(b) 0478 9.4	3.50 // 3.5	**1**	2.00 + 1.00 + 0.50. Ignore a currency symbol. 3 (the count of Textiles jobs) = 0.

4(a)
0478 9.4
	One mark for each correct gap:
	• `FROM RepairJob`
	• `AND`
	• `<`
	• `ORDER BY PartsCost` // `ORDER BY PartsCost ASC` // `ORDER BY PartsCost ASCENDING`
	Completed statement:
	`SELECT Item, PartsCost`
	`FROM RepairJob`
	`WHERE Category = 'Textiles' AND`
	`Minutes < 30`
	`ORDER BY PartsCost ASC;`
	Output (for checking): Curtains 0.50, then Jeans 1.00.
		4	Gap 3: `<=` = 0 (the job must take less than 30 minutes). Gap 4: `ASC` may be left out because ascending is the default. `DESC` = 0. `ORDER BY Minutes` = 0 (wrong field). `OR` in gap 2 = 0.

5(a)
0478 9.4
	One mark for each correct line:
	• `SELECT SUM(Minutes)`
	• `FROM RepairJob`
	• `WHERE Category = 'Bicycle'`
	Example:
	`SELECT SUM(Minutes)`
	`FROM RepairJob`
	`WHERE Category = 'Bicycle';`
	Output (for checking): 65
		3	`COUNT(Minutes)` or `SELECT Minutes` = 0 for the first mark. Accept `SUM (Minutes)` with a space. `Bicycle` written without quotation marks loses the third mark. Extra correct conditions that do not change the result (e.g. `AND Minutes > 0`) are not penalised; `AND Fixed = TRUE` loses the third mark (it leaves out Folding bike). An `ORDER BY` line is ignored.

6(a) 0478 9.4	One mark for each error identified **and** corrected: • Line 01: `SELECT COUNT(JobID)` // `SUM` should be `COUNT` • Line 03: `WHERE Category = 'Electrical'` // quotation marks needed round `Electrical` • Line 04: `AND Fixed = TRUE;` // `OR` should be `AND` Output of the corrected statement (for checking): 3	**3**	Line number and correction are both needed for each mark. Line 01: accept `COUNT(*)` or `COUNT` of any field. Line 04: accept `AND Fixed` // `AND Fixed = Yes`. Line 02 has no error.

Total: 21 marks