ISYS2120 Data and Information Management

THE UNIVERSITY OF SYDNEY · LEVEL 2 · SEMESTER 2 2026

Full guide + ask Sia about ISYS2120

SIDE 1/2 DEFINITIONS → DECISIONS → QUERY LOGIC → CONSTRAINTS → ALGEBRA

CLOSED-BOOK MEMORY SHEET

What must be in your head

0 · Exam Map

COURSE + BRIEF

Final 50 | Quizzes 21 | SQL E | CM 10 | OQ

Final Exam	**50%**	weekly quizzes	21%
SQL tasks	6%	Early feedback quiz	3%
Conceptual Model Assignment	10%	SQL Online Quiz	10%

OFFICIAL RULE

`Paper-based exam, 2 hours, closed book.` *ISYS2120 current Canvas Assessment overview; 2026 intro slides*

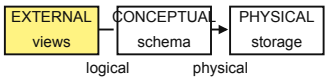
★ Retrieval order: **define the object** → state the formal rule → apply it to the shown rows → name the retained/rejected result → test one boundary case. A plausible diagram or query is incomplete until its meaning and result are checkable.

1 · Database Core

FOUNDATIONS

TERM	RECITE
Data	recordable facts that matter and must persist
Database	persistent organisational data
DBMS	software coordinating storage, metadata, queries, integrity, sharing, recovery and security
Metadata	structure, origin and obligations describing data
Catalogue	database-held names, datatypes and constraints

A relation has uniformly structured tuples over named attributes and domains. Schema is the stable description; instance is the current contents. Every update changes the instance; every valid instance must fit the schema. Mathematical relations have no duplicate rows and no meaningful row order.



independence protects level above

External → Conceptual → Physical; insulation.

RECALL → *Logical independence protects user views; physical independence protects the conceptual design from storage changes.*

2 · Keys & Integrity

FOUNDATIONS

CANDIDATE-KEY TEST

`Proposed columns distinguish every row in` **`every permitted state`**`, not merely the sample; one counterexample rejects the claim.`

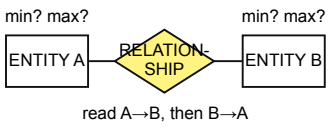
Domain/type	permitted value kind
PRIMARY KEY	selected identity: unique and present
FOREIGN KEY	non-NULL value matches a target key

4 · ER Constraint Decoder

ER MODELLING

Start from requirements. An ***entity type*** groups distinguishable objects; an ***attribute*** describes an entity or relationship; a ***relationship type*** records a fact among roles. Ask whether each noun needs independent identity and retained facts.

Composite attribute	has meaningful parts
Multivalued attribute	one entity may have several values
Relationship attribute	depends on the participant combination



read A→B, then B→A

Read maximum and minimum at both ends.

QUESTION	CONSTRAINT	MEANING
How many at most?	maximum cardinality / key constraint	limits counterparts
Must every one participate?	minimum / total participation	at least one relationship instance
Who does what?	role name	disambiguates recursive participation

Say every mark in plain domain language. Then narrate one valid and one invalid population. If requirements are silent, state the assumption instead of smuggling in a business rule.

WEAK ENTITY

A weak entity's identity combines an owner's key with a local discriminator. Few attributes or low importance do not make it weak; **owner-dependent identity** does.

A key is minimal: remove a necessary component and it no longer distinguishes every entity.



identity = owner key + discriminator

Owner key + discriminator identifies the weak entity.

HARDER STRUCTURES

Recursive	same entity type plays distinct named roles
ISA	choose subtype constraints deliberately; preserve inheritance
Ternary / aggregation	keep the combined fact when binary links would lose or invent combinations

5 · SQL: Build the Result

QUERY FOUNDATIONS



rows → groups → output → presentation

plan meaning before clause syntax

Plan rows → groups → output; sort last.

CLAUSE / IDEA	JOB	DIAGNOSTIC
SELECT	target columns / expressions	only requested output?
FROM	source relations	all needed, no extras?
JOIN / ON	connect matching facts	missing predicate multiplies rows
WHERE	keep rows whose predicate is TRUE	boundary and NULL case?
GROUP BY	partition surviving rows	right population?
HAVING	filter completed groups	aggregate belongs here
ORDER BY	sort result	membership unchanged

FAST QUERY PLAN

`OUTPUT columns → SOURCE relations → CONNECTIONS → FILTERS → predicted smallest result.`

NULL IS UNKNOWN

Comparisons with NULL do not become TRUE; use `IS NULL`, not `= NULL`. Keep missing values explicit when predicting filters and aggregates. SQL normally preserves duplicate result rows; `DISTINCT` removes duplicates only when uniqueness is required.

TRAP → *DISTINCT is not a repair for a faulty join. First explain why repeated rows exist; then decide whether the requested projected values should be unique.*

5b · SQL Contrast Deck

QUERY FOUNDATIONS

PAIR	DIFFERENCE TO RECITE
projection / selection	output attributes versus retained rows
equi-join / natural join	explicit equality keeps both match columns; natural matches every same-named column and keeps one copy
alias / stored name	temporary scope name versus schema object name
duplicate rows / repeated facts	same projection need not mean the source fact is duplicated
membership / order	WHERE decides inclusion; ORDER BY changes presentation
expression / stored value	target expression computes output without adding a base column

6 · Joins, Groups, Tests

SQL + ADVANCED

IF THE REQUEST SAYS…	USE / PROTECT
only matching facts	inner / theta join with explicit condition
every left row, even none	LEFT JOIN; unmatched right attributes become NULL
one table in two roles	aliases; qualify attributes
summary per value	GROUP BY that value
filter raw rows	WHERE before grouping
filter summaries	HAVING after grouping

KEEP EMPTY?

LEFT JOIN loans ON key AND open

YES: filter in ON

WHERE r.open rejects NULL rows

NO: rows vanish

Put right filters in ON to retain empty left rows.

COUNTING RETAINED LEFT ROWS

`Start from every left entity → LEFT JOIN matches → GROUP BY left key → COUNT(non-NULL right key). COUNT(*) counts the joined row, including an unmatched NULL-extended row.`

DIAGNOSTIC DATA

ordinary match	base meaning
unmatched row	inner vs outer join
repeated projected value	bag vs DISTINCT
NULL	three-valued filter / aggregate
boundary value	> versus ≥
empty group	retention and count choice

Predict intermediate rows by hand. The first stage where prediction and result diverge identifies the faulty clause. An accidental Cartesian product pairs every row from one source with every row from the other and can make credible-looking but inflated summaries.

EXPLAIN → *State why every returned row satisfies the request and why one boundary row is excluded. A query that runs is not yet evidence that the meaning is right.*

6b · Grouping Logic

ADVANCED SQL

LOGICAL POPULATION

`FROM/JOIN creates candidates → WHERE removes rows → GROUP BY partitions survivors → aggregates summarise each group → HAVING removes groups → SELECT forms output.`

COUNT(*)	counts retained joined rows
COUNT(column)	ignores NULL in that column
Unmatched left row	one NULL-extended row survives a LEFT JOIN
Empty-match count	count a right-side key to obtain zero

7 · DDL as Contract

SCHEMA DESIGN



validate the complete post-change state

Each change is tested against every active rule.

DDL DEVICE	PROMISE	FAILING PROBE
Datatype	column value kind	incompatible value
NOT NULL	value present	missing mandatory field
PRIMARY KEY	unique, non-NULL identity	duplicate full key
UNIQUE	distinct non-NULL value/combination	repeated candidate value
FOREIGN KEY	matching referenced key	dangling non-NULL reference
CHECK	row satisfies Boolean rule	boundary-breaking value

SQL permits multiple NULLs in a UNIQUE column unless it is also NOT NULL. A composite primary key constrains the **combination**; individual components may repeat. Referential actions are domain policies: restriction blocks a change; cascading propagates it; setting NULL needs an optional reference. Historical retention may rule out automatic deletion.

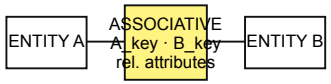
ANSWER SCAFFOLD

`Requirement → declared type/constraint → attempted invalid row → named rejection → resulting stored table.`

8 · ER → Relations

SCHEMA MAPPING

CONCEPTUAL PATTERN	RELATIONAL DECISION
strong entity	relation; entity key becomes relation key
simple attribute	column
composite attribute	flatten into component columns
multivalued attribute	separate relation
1:M relationship	one-side key migrates as FK to many side
M:N relationship	associative relation with both FKs; relationship attributes live there
weak entity	owner key + discriminator form identity
total participation	use NOT NULL where the mapping can enforce it



M:N fact becomes its own relation

M:N becomes a relation holding both keys.

9 · Relational Algebra

ALGEBRA + SQL

OPERATOR	EFFECT	CHECK
$\sigma_p(R)$	selection: rows satisfying p	heading preserved
$\pi_A(R)$	projection: named attributes	duplicates removed under set semantics
$\rho(R)$	rename relation/attributes	scope unambiguous; enables self-join
$R \times S$	all tuple combinations	usually constrain into join
$R \bowtie_p S$	pairs satisfying p	meaningful match
$\cup, \cap, -$	set union/intersection/difference	union-compatible headings/domains
$R \div S$	candidates paired with every S tuple	complete divisor audited

TRANSLATION WARNING

`SQL SELECT ≈ algebraic` **`projection π;`** `SQL WHERE ≈ algebraic` **`selection σ.`**



label every intermediate heading + tuples

Every operator returns another relation.

Push selection before expensive combinations when equivalent. For set operators, matching column counts are insufficient: corresponding attributes need compatible domains *and meanings*. Equivalence means the same relation for every valid database state under the stated set/bag and NULL semantics, not agreement on one convenient dataset.

10 · "Every" / Division

ADVANCED SQL

KEEP CANDIDATE c IFF

NO required r lacks a matching fact(c,r)

division ⇔ double NOT EXISTS

"Every" means no required item is missing.

DOUBLE NEGATION

```
SELECT DISTINCT c.id FROM Candidate c
WHERE NOT EXISTS (
 SELECT 1 FROM Required r
 WHERE NOT EXISTS (
  SELECT 1 FROM Fact f
  WHERE f.id=c.id AND f.item=r.item));
```

1. Write the complete divisor / required list.
2. For each candidate, tick every required pairing.
3. One missing pair removes the candidate.
4. Extra pairings do not disqualify it.
5. Preserve both correlations; dropping one can reverse the meaning or test a global condition.

Your exam is closed book: nothing printed goes in. Use this sheet to revise and self-test, not to carry.
Semester 2 formal examination period, from 16 November 2026 (the Exams Office publishes the exact date).

ISYS2120 Worked Recall · Judgement Cards · Exam-Morning Plan

DATA AND INFORMATION MANAGEMENT · CLOSED-BOOK SELF-TEST SIDE

Full guide + ask Sia about ISYS2120

SIDE 2/2 · 5 WORKED PATHS · IF X → Y CARDS · CLASSIC TRAPS · MORNING RETRIEVAL

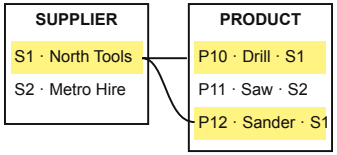
ANSWER BEFORE REVEALING THE PATH

Use paper; no software

11 · Worked: Linked Facts

FOUNDATIONS · SOURCE VALUES

Given. Supplier: (S1, North Tools), (S2, Metro Hire). Product: (P10, Drill, S1), (P11, Saw, S2), (P12, Sander, S1). Return products supplied by North Tools.



Follow S1 → keep P10 Drill and P12 Sander.

1. Find North Tools in Supplier → linking value **S1**.
2. Keep Product rows whose supplier_id matches S1.
3. Project requested product columns; preserve both matches.
4. Result: `P10 | Drill; P12 | Sander`. P11 is excluded because S2 identifies Metro Hire.

TRAP → *Do not match on supplier name when the schema links through supplier_id.*

12 · Worked: Filter Trips

SQL FOUNDATIONS · SOURCE VALUES

Trip(trip_id, depart_date, capacity, booked): T1=(2026-09-09,4,4), T2=(2026-09-10,6,4), T3=(2026-09-11,3,3), T4=(2026-09-12,8,5). Return trips on/after 2026-09-10 with spare seats.

```
SELECT trip_id, capacity - booked AS spare_seats
FROM Trip
WHERE depart_date ⩾ '2026-09-10'
 AND capacity > booked
ORDER BY trip_id;
```

ROW	DATE?	SPARE?	OUTPUT
T1	no	—	exclude
T2	yes	6−4	**2**
T3	yes	3>3 false	exclude
T4	yes	8−5	**3**

Answer: T2 | 2; T4 | 3. Decide membership before computing the target expression.

12b · Re-run Without Looking

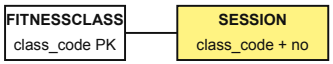
LINKED ROWS + SELECT

PROMPT	ANSWER PATH
Why do P10 and P12 both remain?	both Product tuples carry S1; sharing a supplier does not merge distinct products
Why exclude P11?	its linking value S2 identifies Metro Hire
What if you project only supplier_id?	repeated S1 values remain in SQL unless DISTINCT is requested
Which Trip predicate removes T1?	date threshold
Which removes T3?	capacity > booked is false at equality

13 · Worked: Weak Session

ER MODELLING · SOURCE VALUES

A FitnessClass is identified by class_code. Session numbering restarts inside each class; a Session cannot exist without its class and has start_time. For YOGA: session 1 at 09:00; session 2 at 17:00.



Session(code PK/FK, no PK, start)

YOGA · 1 · 09:00

YOGA · 2 · 17:00

Owner key + discriminator identifies Session.

1. Owner: FitnessClass with class_code key.
2. Local discriminator: session_no; not globally unique.
3. Map Session(class_code, session_no, start_time); composite PK is **(class_code, session_no)**, and class_code is also the owner FK.
4. Another class may validly have session_no 1.

TRAP → *session_no alone would wrongly prevent PILATES | 1; omitting class_code loses the identifying owner.*

14 · Worked: Composite DDL

SCHEMA DESIGN · SOURCE VALUES

WorkshopSession(event_code, sequence_no, room, capacity): the first two columns identify a row; capacity must be positive.

```
CREATE TABLE WorkshopSession(
 event_code TEXT NOT NULL,
 sequence_no INTEGER NOT NULL,
 room TEXT,
 capacity INTEGER CHECK(capacity > 0),
 PRIMARY KEY(event_code, sequence_no));
```

ATTEMPT	RESULT
E1,1,R2,20	store
E1,2,R3,15	store
E2,1,R2,10	store
E1,1,R9,30	**reject duplicate pair**

Repeated E1 and repeated sequence_no 1 are individually valid. The full pair (E1,1) cannot repeat; the failed row is not stored.

14b · Constraint Attack Deck

DDL SELF-TEST

ATTACK	EXPECTED EXPLANATION
omit event_code	NOT NULL / primary-key identity cannot be missing
repeat E1 with sequence 3	valid: the full pair is new
use E3 with sequence 1	valid under the shown table: sequence_no is not globally unique
capacity at or below zero	CHECK(capacity > 0) rejects the row

15 · Worked: Division

ADVANCED SQL · SOURCE VALUES

ToolStock: (D1,Hammer), (D1,Drill), (D2,Hammer), (D3,Hammer), (D3,Drill), (D3,Saw). **Required**: Hammer, Drill. Which depots stock every required tool?

depot	Hammer	Drill	extra
D1	✓	✓	—
D2	✓	✗	—
D3	✓	✓	Saw

quotient = {D1, D3}

D1,D3 pass; D2 lacks Drill; extra Saw is fine.

1. Divisor contains Hammer and Drill.
2. D1 has both; retain. D2 lacks Drill; remove. D3 has both; retain.
3. ToolStock ÷ Required = {D1, D3}.
4. Equivalent SQL uses correlated double NOT EXISTS; compare its ordered output with the hand quotient.

```
SELECT DISTINCT t.depot_id FROM ToolStock t
WHERE NOT EXISTS (SELECT 1 FROM Required r
 WHERE NOT EXISTS (SELECT 1 FROM ToolStock x
 WHERE x.depot_id=t.depot_id AND x.tool=r.tool))
ORDER BY t.depot_id;
```

16 · Five Answer Paths

FAST RECALL

Linked facts	lookup key → match FK → project
SELECT	sources → filters → expression → order
Weak entity	owner → discriminator → composite identity
DDL	requirement → constraint → violating row → stored state
Division	divisor list → candidate ticks → quotient

CLOSED-BOOK DRILL → *Cover the worked path; reproduce the result and one rejected row. Then explain the formal rule without looking.*

16b · Algebra Micro-Drills

OPERATOR TRACE

PROMPT	WHAT TO WRITE FIRST
keep open loans	$\sigma_{status='open'}$(Loan); same heading, fewer tuples
return memberId only	$\pi_{memberId}(\ldots)$; name output heading and remove duplicate tuples under set semantics
compare two bookings	ρ creates two scopes before the self-join
combine then match	product creates all pairs; theta-join adds the meaningful predicate

17 · If You See X → Do Y

METHOD CHOOSER

SEE	DO
every / all required	÷ / 2×NOT EXISTS
include zero rows	LEFT JOIN; count key
number per owner	weak = owner + no.

Trigger word → method; verify domain meaning.

SEE	DO
"how many at most?"	maximum cardinality in both directions
"must participate?"	minimum / total participation
same entity twice	role names + aliases / rename
raw row condition	WHERE
aggregate condition	HAVING
unique output values	DISTINCT, after join is proven
missing value	IS NULL; predict UNKNOWN comparisons
many-to-many	associative relation with both keys
invalid history deletion	restrict; do not assume CASCADE
set union/difference	check headings, domains and meanings

UNIVERSAL SENTENCE

"There does not exist a required item for which there does not exist a matching fact."

18 · Result-Prediction Grid

QUERY VERIFICATION

ADD THIS ROW	IT EXPOSES	WRONG SYMPTOM
unmatched parent	outer join	row disappears
duplicate projected value	bag semantics	unexpected repetition
NULL optional value	unknown / aggregates	row silently filtered
boundary equality	> vs ⩾	edge membership wrong
one missing required pair	division	false universal pass
duplicate composite pair	key scope	invalid row accepted

BEST EVIDENCE → *The smallest counterexample that separates your intended result from a plausible wrong result.*

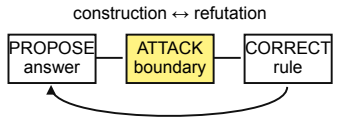
18b · Modelling Judgement Cards

ER SELF-TEST

QUESTION	DECISION TEST
Entity or attribute?	Does it need independent identity, retained facts and participation in relationships?

19 · Classic Trap Board

ALL FIVE CHAPTERS



Propose → attack a boundary → correct the rule.

WRONG MOVE	CORRECTION TO RECITE
sample rows prove a key	key must distinguish every permitted state
cardinality = participation	maximum and minimum are separate
few attributes ⇒ weak	weak means owner-dependent identity
`= NULL`	use IS NULL; NULL comparison is unknown
two tables, no join predicate	Cartesian product; connect matching values
DISTINCT hides extra rows	repair join first; deduplicate only by meaning
aggregate in WHERE	WHERE rows; HAVING groups
LEFT JOIN then right filter in WHERE	put qualifying right filter in ON if unmatched left rows survive
COUNT(*) for optional matches	count a non-NULL right key
separate UNIQUE columns for pair	one composite constraint on the pair
M:N foreign key on one side	associative relation
SQL SELECT = algebraic selection	SQL SELECT maps to projection π
one match proves "every"	audit all divisor items; seek missing pair
same output on one dataset proves equivalence	must agree on every valid state under stated semantics

CORRECTION LANGUAGE → *"This fails when …; the rule is …; therefore the result must …". That sentence forces a counterexample, a formal principle and an observable consequence.*

19b · Definition Flash Grid

FAST RECALL

TERM	DISTINGUISHING SENTENCE
schema / instance	stable structure versus contents at one time
relation / table output	set of tuples with named attributes; mathematical relation has no order or duplicates
candidate / primary key	any row-distinguishing combination versus the selected declared identifier
foreign key	non-NULL reference must match a target key
total participation	every entity must occur in at least one relationship instance
selection / projection	retain rows versus retain attributes
WHERE / HAVING	filter candidate rows versus completed groups

20 · Exam-Morning Plan

CLOSED-BOOK RETRIEVAL



Define → draw → trace → attack, without looking.

LAST REVISION SESSION

1. From memory, contrast data/database/DBMS; schema/instance; candidate/primary/foreign key; SQL/algebraic selection/projection.
2. Draw External → Conceptual → Physical and name logical versus physical independence.
3. Draw one ER relationship; read maximum and minimum both ways. Add a weak entity and explain its composite identity.
4. Write one SELECT with a join, WHERE, GROUP BY and HAVING. Predict the intermediate rows.
5. Write one CREATE TABLE with a composite key, FK and CHECK; invent a failing row for each.
6. Compute the ToolStock division and narrate double NOT EXISTS.

ON THE PAPER

Definitions	term → distinguishing contrast → consequence
ER	requirements → identity → max/min both ways → counterexample
SQL	output → sources → joins → filters → result rows
DDL	rule → syntax → invalid state rejected
Algebra	heading + tuples after every operator
Universal	complete required list; find missing pair

Never stop at a name. A full answer says what the object is, what rule applies, what happens to the supplied rows, and which counterexample would falsify the proposed answer.

21 · 60-Second Oral Check

FAST RECALL

Why a DBMS?	shared persistent data + catalogue + integrity + controlled services
What proves a key?	uniqueness over all permitted states
Why weak?	identity depends on owner + discriminator
WHERE vs HAVING?	rows before grouping vs completed groups
Why LEFT JOIN?	retain unmatched left rows with NULL right side
Why division?	candidate related to every required tuple

FINISH → *If formal and domain explanations disagree, revisit the assumption before practising another problem.*

Your exam is closed book: nothing printed goes in. Use this sheet to revise and self-test, not to carry.
Semester 2 formal examination period, from 16 November 2026 (the Exams Office publishes the exact date).