

TD No. 1 — Introduction to Advanced Databases

Exercise 1 — Mark True or False (T/F) for each statement — Briefly justify your answer

#	Statement	T/F	Justification
1	The schema changes frequently, the instance is stable.		
2	A primary key can be null.		
3	DDL deals with structure, DML with data.		
4	ACID concerns transaction reliability.		
5	The E/R model is identical to the relational model.		
6	In ANSI/SPARC, the external level refers to user views.		
7	MVCC increases the parallelism of reads.		
8	NoSQL databases always replace RDBMSs.		
9	A foreign key must reference an existing primary key.		
10	An index changes the result of a SELECT query.		

Exercise 2 — Match each Number to its Corresponding Definition

No.	Abbreviation / Concept	Order	Definition
1	DDL	M	Table of the relational model (set of tuples and attributes).
2	DML	B	User view (representations adapted to needs).
3	DCL	P	Descriptive property of an entity/association.
4	TCL	G	Software managing storage, integrity, access, and optimization (DBMS).
5	Referential integrity	C	Manipulation queries: SELECT, INSERT, UPDATE, DELETE.
6	External level (ANSI/SPARC)	J	Structure definition: CREATE, ALTER, DROP.
7	Candidate key	D	Schema / Instance: stable structural plan vs. data at a given time.
8	PK (Primary Key)	H	Conceptual link between entities (Entity–Relationship model).
9	FK (Foreign Key)	A	Uniqueness constraint identifying each tuple.
10	Schema / Instance	N	Transactional control: COMMIT, ROLLBACK, SAVEPOINT.
11	ACID	E	Access control commands: GRANT, REVOKE.
12	Entity	O	Reference required to an existing key in another table.
13	Attribute	L	Set of columns that can serve as a primary key.
14	Relation	F	Transaction reliability properties (ACID).
15	Association (Entity–Relationship model)	I	Constraint pointing to an existing key in another table.
16	DBMS	K	Object or concept from the real world being modeled.

Exercise 3 — Data Integrity

We consider the following **fictitious schema** for a database named **“Bank Account Management”**:

AGENCY(CODE, ADDRESS)

ACCOUNT(CODE, NUMBER, CLIENT, ADDRESS)

TRANSACTION(CODE, NUMBER, TIMESTAMP, OPERATION, AMOUNT, BRANCH_BALANCE, CLIENT_BALANCE)

Sample Instance (for context):

AGENCY

CODE	ADDRESS
2101	CENTRE VILLE SKIKDA
2512	CONSTANTINE CENTRE

ACCOUNT

CODE		NUMBER	CLIENT	ADDRESS
2512		2426136-90	CHIEKH NASS MLAH	4 RUE SAINT JEAN, 25000 CONSTANTINE
2101		2425368-11	TAHAR OULD TAHAR	2 RUE ALLÉES, 21000 SKIKDA
2512		5252636-20	ALI BEN ALI	ROUTE BAB EL OUED No 5, 25200 BELVUE

TRANSACTION

CODE	NUMBER	TIMESTAMP	OPERATION	AMOUNT	BRANCH_BALANCE	CLIENT_BALANCE
2101	2426136-90	SYS_TS	DEPOSIT	15000.00	?	?
2101	2425368-11	SYS_TS	DEPOSIT	10000.00	?	?
2512	2426136-90	SYS_TS	WITHDRAWAL	5000.00	?	?
2101	5252636-20	SYS_TS	DEPOSIT	12000.00	?	?
2512	2426136-90	SYS_TS	WITHDRAWAL	12000.00	?	?
2512	5252636-20	SYS_TS	DEPOSIT	12000.00	?	?

(TIMESTAMP takes the system timestamp value.)

We want to **design this database** while ensuring **data integrity** through appropriate structural constraints.

1) Domain Constraints (DOMAIN)

Propose domain types (or equivalent) for the following:

Symbol	Specification
DCODE	Agency code format AABB, with AA ∈ [01..48] and BB ∈ [01..99]. Include pattern check, NOT NULL if relevant, and optional collation.
DTEXT	Alphanumeric strings (≈ 60 characters max) for CLIENT and ADDRESS. Specify length, UPPER/TRIM if needed, NOT NULL if relevant.
DNUMBER	Account number format CCCCCC-CC (only digits). Define pattern/length and NOT NULL if relevant.
DMONETARY	Numeric type for amounts/balances. Specify range, CHECK (value ≥ 0), NOT NULL if relevant.
DOPERATION	Controlled values: 'DEPOSIT' or 'WITHDRAWAL'. Specify enumeration, case sensitivity, NOT NULL.
DTIMESTAMP	System timestamp type (+ appropriate DEFAULT). Specify NOT NULL if relevant, and prohibit future dates.

2) Structural Constraints (Schema)

Declare and justify **PK, FK, UNIQUE, NOT NULL, CHECK, DEFAULT** and **ON DELETE/ON UPDATE** actions.

- **AGENCY**
 - CODE: PK, type DCODE.
 - ADDRESS: DTEXT, NOT NULL.
- **ACCOUNT**
 - PK = (CODE, NUMBER) with CODE: DCODE, NUMBER: DNUMBER.
 - CODE: FK → AGENCY(CODE); specify reference action and justify (choice between CASCADE / RESTRICT / SET NULL).
 - CLIENT, ADDRESS: DTEXT, NOT NULL.
 - Optional UNIQUE(NUMBER) if account numbers are globally unique — justify.
- **TRANSACTION**
 - FK (CODE, NUMBER) → ACCOUNT(CODE, NUMBER) with reference actions and justification.
 - CODE: FK → AGENCY(CODE) for agency consistency.
 - TIMESTAMP: DTIMESTAMP with system DEFAULT.
 - OPERATION: DOPERATION with CHECK on valid values.
 - AMOUNT, BRANCH_BALANCE, CLIENT_BALANCE: DMONETARY, NOT NULL, CHECK (≥ 0).

3) Tuple-Level CHECK Constraints — Table TRANSACTION

Specify at least two CHECK constraints:

1. Date validation: prohibit future timestamps (and/or invalid formats).
2. Amount/Type consistency: if OPERATION = 'WITHDRAWAL' then AMOUNT > 0.

Explain why a CHECK on a nullable column must tolerate NULL (e.g., VALUE IS NULL OR condition) — because SQL treats NULL as “unknown”, so the constraint must not reject tuples when the tested value is missing.

4) Non-Structural Constraints (Business Logic)

List 3–5 business rules not purely structural and explain how to enforce them (using **triggers**, **secure views**, or **stored procedures**).

Examples include:

- Daily withdrawal limit per account.
- Account balance must never be negative after a withdrawal.
- Agency consistency: TRANSACTION.CODE must equal the CODE of the referenced account.
- Detection of duplicate transactions at the same instant (same account + identical rounded timestamp).