

1. ACID Properties of Transactions

ACID ensures reliable database transactions:

1. **Atomicity:** A transaction is *all or nothing*. Either the entire transaction is completed, or none of it occurs.
 2. **Consistency:** Ensures the database moves from one valid state to another, maintaining all constraints (e.g., primary keys, foreign keys).
 3. **Isolation:** Transactions are processed independently, so one transaction's changes don't affect another until it's complete.
 4. **Durability:** Once a transaction is committed, its changes are permanent, even if the system crashes.
-

2. Isolation Levels and Dirty Read Example

Isolation levels determine how transactions interact with each other.

1. **Read Uncommitted:** Allows *dirty reads* (reading uncommitted data).
Example: In an attendance system, if someone's arrival is updated but not committed, other users might see inconsistent counts.
2. **Read Committed:** Prevents dirty reads; only committed data can be read.
Example: The attendance count will reflect only completed updates.
3. **Repeatable Read:** Ensures no data changes during a transaction. However, new rows (phantoms) can still appear.
Example: If you calculate attendance, no student count changes during your calculation.
4. **Serializable:** Highest isolation, prevents dirty reads, non-repeatable reads, and phantoms by locking rows or tables.
Example: Attendance calculation is entirely locked until the transaction finishes.

Two Ways to Prevent Dirty Reads:

1. **Row Locks:** Lock the rows being modified to prevent concurrent access.
 2. **Snapshot Isolation:** Creates a copy of data at the start of a transaction.
-

3. Normalization

Normalization organizes a database to reduce redundancy and improve consistency:

1. **1NF (First Normal Form):** Ensures each column has atomic (indivisible) values and unique rows.
Example: A table storing names as separate columns (First Name, Last Name).
2. **2NF (Second Normal Form):** Removes partial dependencies. Each non-key column depends entirely on the primary key.
Example: In a student table, avoid repeating course details.

3. **3NF (Third Normal Form):** Removes transitive dependencies (non-key columns shouldn't depend on other non-key columns).
Example: Separate student details from department information.
 4. **BCNF (Boyce-Codd Normal Form):** A stricter version of 3NF. Every determinant (column defining another) must be a candidate key.
Example: Resolves edge cases where non-prime attributes define candidate keys.
-

4. Docker Basics

- **Starting with Docker:** Install Docker, then use commands to manage containers.
Example:
 - `docker ps`: Lists running containers.
 - `docker ps -a`: Lists all containers (running + stopped).
 - **Entering a MySQL Container:**
Command: `docker exec -it <container_id> bash`
Example: `docker exec -it 0a bash` to access the MySQL editor.
-

5. Taking a Database Backup

Command:

```
mysqldump -u root -p <database_name> > /path/to/backup.sql
```

Example:

```
mysqldump -u root -p kaushal > /home/backup.sql  
Password: admin
```

Explanation: This creates a dump of the `kaushal` database and saves it as `backup.sql` in `/home`.

6. Adding an Index

Command:

```
ALTER TABLE people ADD INDEX birthday_index(birthday);
```

Explanation:

This adds an **index** named `birthday_index` on the `birthday` column in the `people` table, improving search performance for birthday queries.