

# ■ Database Interaction Checklist for Backend Developers

## ■ Security

- ■ Use parameterized queries / ORM placeholders (no string concatenation).
- ■ Sanitize and validate all user inputs.
- ■ Store DB credentials in environment variables, not in code.
- ■ Use least privilege for the DB user (no DROP/ALTER if not needed).
- ■ Hash passwords with bcrypt/argon2 (never store plaintext).
- ■ Encrypt sensitive fields if required (PII, financial info).

## ■ Performance

- ■ Select only the required fields (no SELECT \*).
- ■ Check if indexes exist on frequently queried columns.
- ■ Avoid N+1 query problem (use eager loading/joins).
- ■ Use pagination for large result sets.
- ■ Consider caching (Redis, Memcached) for heavy reads.
- ■ Batch inserts/updates where possible.

## ■ Transactions & Consistency

- ■ Wrap multi-step operations in a transaction.
- ■ Ensure proper rollback on failure.
- ■ Think about isolation levels if concurrent writes happen.
- ■ Validate data before insert/update to avoid constraint violations.

## ■ Error Handling

- ■ Catch and handle DB errors gracefully (timeouts, duplicate keys, deadlocks).
- ■ Add retry logic for transient errors.
- ■ Log query errors with enough context (user, params, etc.).
- ■ Return meaningful error messages to API clients (not raw SQL errors).

## ■ Scalability

- ■ Use a connection pool (avoid creating connections per request).
- ■ Profile queries with EXPLAIN or ORM logging if performance issues.
- ■ Offload heavy queries to background jobs/queues.
- ■ Plan for sharding or replication if dataset grows large.

## ■ Maintainability

- ■ Apply migrations for schema changes (don't change DB manually).
- ■ Use consistent naming conventions (snake\_case, camelCase, etc.).
- ■ Centralize DB logic in repository/service layers.
- ■ Write unit & integration tests for DB interactions.
- ■ Document schema relationships (ER diagrams, README notes).

## ■ Data Integrity

- ■ Use foreign keys, unique, not null constraints in DB.
- ■ Pick correct data types (DATE, BOOLEAN, etc. not just TEXT).
- ■ Apply default values where applicable.
- ■ Log or audit critical changes (salary, permissions, roles).

## ■ Monitoring & Maintenance

- ■ Enable slow query logging.
- ■ Monitor DB connections, latency, memory.
- ■ Set up backups and test restoring them.
- ■ Regularly review queries and indexes.