Solutions

2.1

- The WHERE becomes always-true: all 8 students match ✓.

2.2

- The compiled query text; the bound parameter values ✓.

2.3

- Every application bug would run with full owner authority – no bounds on the blast radius ✓.

3.1

- `GRANT SELECT ON Student TO app; GRANT SELECT ON Course TO app;` ✓.
- `GRANT SELECT, INSERT, DELETE ON Enrollment TO app;` ✓.
- Stolen session: can read students and courses, enroll and un-enroll – cannot touch Department, cannot drop or alter anything ✓.
- Auditor: `GRANT SELECT ON access_log TO auditor;` ✓.
- Revoke matters because privileges creep: old roles accumulate rights nobody uses – the audit loop must shrink them ✓.

3.2

- The query is compiled with placeholders: the attacker's text is only ever bound as a value compared for equality – it cannot become SQL ✓.
- Governance: classify the data, assign owners, set retention, audit access ✓.
- Encryption protects data at rest – a valid login reads through the database's own decryption; grants bound what the login may select ✓.