Chapter 7: Python with Databases (Page 109)

:Multiple Choice (MCQs)

1. sqlite3 (c
2. Saves changes permanently in the database (Section 7.3) (b.
3. (Section 7.4)؟ (c.
4. fetchone() (Section 7.5) (c.
5. Base (Section 7.9 questions imply SQLAlchemy knowledge, (a typically DeclarativeBase).

:True/False

1. (SQLite stores data in a file on disk) **False**.
2. (Parameterized queries prevent injection) **True**.
3. (rollback undoes uncommitted changes) **True**.
4. (SQLAlchemy has both Core and ORM) **True**.

False (It returns a cursor object; results are fetched via methods like fetchall).

```
# ========================================
# Chapter 7: Databases
# ========================================
print("\n--- Chapter 7 Solutions ---")

conn = sqlite3.connect(':memory:')
cursor = conn.cursor()

# 1. Basic CRUD
cursor.execute("CREATE TABLE students (id INTEGER PRIMARY KEY, name TEXT,
grade REAL)")
cursor.executemany("INSERT INTO students (name, grade) VALUES (?, ?)",[('Ali',
85.5), ('Sara', 92.0), ('Mohamed', 78.3)])
```

```
cursor.execute("SELECT * FROM students")
print(f"All Students: {cursor.fetchall()}")

# 2. Parameterized Queries
name, grade = "Amina", 88.5
cursor.execute("INSERT INTO students (name, grade) VALUES (?, ?)", (name,
grade))
cursor.execute("SELECT * FROM students")
print(f"Updated Records: {cursor.fetchall()}")
conn.commit()
# 3. Transactions
try:
    conn.execute("BEGIN")
    cursor.execute("INSERT INTO students (name, grade) VALUES ('Temp', 50)")
    raise ZeroDivisionError
    conn.commit()
except ZeroDivisionError:
    conn.rollback()
    print("Transaction rolled back due to error.")

conn.close()
```