**CSC 179 Database Transaction Control Flow Nodes**

1. InsertData(Str, Str)

public boolean insertData(String name, String desc){  
 SQLiteDatabase db = this.getWritableDatabase();  
 Gson gson = new Gson();  
  
 ArrayList<Long> points = new ArrayList<>();  
 points.add(0L);  
  
 if(!name.isEmpty()) {  
 if (desc.isEmpty()) desc = "No Description";  
 ContentValues cv = new ContentValues();  
 cv.put(*COLUMN\_NAME*, name);  
 cv.put(*COLUMN\_TIME*, 0);  
 cv.put(*COLUMN\_DESC*, desc);  
 cv.put(*COLUMN\_POINTS*, gson.toJson(points, t));  
  
 return db.insert(*TABLE\_NAME*, null, cv) != -1;  
 }  
 else return false;  
}

**Basic Blocks (nodes)**

1. db = this.getWriteableDatabase();

gson = new Gson();

points = new ArrayList<>();

points.add(0L);

1. if(!name.isEmpty())
2. if(desc.isEmpty())
3. desc = “No Description”;
4. cv = new ContentValues();

cv.put(COLUMN\_NAME, name);

cv.put(COLUMN TIME, 0);

cv.put(COLUMN\_DESC, desc);

cv.put(COLUMN POINTS, gson.toJson(points, t));

1. return db.insert(TABLE\_NAME, null, cv) != -1;
2. else return false

**Entry Node(s)**

1

**Exit Node(s)**

6,7

**Control Flow**

1->2

2->3, 2->7

3->4, 3->5

4->5

5->6

1. updateData(Str, Str, Str)

public boolean updateData(String oldName, String newName, String desc){  
 SQLiteDatabase db = this.getWritableDatabase();  
  
 if(!newName.isEmpty()) {  
 if(desc.isEmpty()) desc = "No Description";  
 ContentValues cv = new ContentValues();  
 cv.put(*COLUMN\_NAME*, newName);  
 cv.put(*COLUMN\_DESC*, desc);  
  
 return db.update(*TABLE\_NAME*, cv, "NAME = ?", new String[]{oldName}) != -1;  
 }  
 else return false;  
}

**Basic Blocks**

1. db = this.getWriteableDatabase();
2. if(!newName.isEmpty())
3. if(desc.isEmpty())
4. desc = “No Description”;
5. cv = new ContentValue();

cv.put(COLUMN\_NAME, newname);

cv.put(COLUMN\_DESC, desc);

1. return db.update(TABLE\_NAME, cv, “NAME = ?“, new String[]{oldName}) != -1;
2. return false

**Entry Node(s)**

1

**Exit Node(s)**

6,7

**Control Flow**

1->2

2->3, 2->7

3->4, 3->5

4->5

5->6

1. deleteData(str)

public boolean deleteData(String name){  
 SQLiteDatabase db = this.getWritableDatabase();  
 return db.delete(*TABLE\_NAME*, "NAME = ?", new String[] {name}) !=-1;  
}

**Basic Blocks**

1. db = this.getWriteableDatabase();
2. return db.delete(TABLE\_NAME, “Name = ?”, new String[] {name}) != -1;

**Entry Node(s)**

1

**Exit Node(s)**

2

**Control Flow**

1->2

1. getTasks()

public ArrayList<String> getTasks(){  
 SQLiteDatabase db = this.getWritableDatabase();  
 ArrayList<String> tasks = new ArrayList<>();  
  
 Cursor res = db.rawQuery("SELECT "+ *COLUMN\_NAME*+" FROM "+ *TABLE\_NAME*, null);  
  
 for(res.moveToFirst();!res.isAfterLast();res.moveToNext()){  
 tasks.add(res.getString(0));  
 }  
  
 return tasks;  
}

**Basic Blocks**

1. db = this.getWriteableDatabase();

tasks = new ArrayList<>();

res = db.rawQuery(“SELECT “ + COLUMN\_NAME + “ FROM “ + TABLE\_NAME, null);

res.moveToFirst();

1. !res.isAfterLast();
2. tasks.add(res.getString(0));
3. res.moveToNext();
4. return tasks;

**Entry Node(s)**

1

**Exit Node(s)**

5

**Control Flow**

1->2

2->3, 2->4, 2->5

3->2

4->2

1. getAllData()

public ArrayList<Task> getAllData(){  
 SQLiteDatabase db = this.getWritableDatabase();  
 ArrayList<Task> tasks = new ArrayList<>();  
 Cursor cursor = db.rawQuery("SELECT \* FROM "+ *TABLE\_NAME*, null);  
  
 if(cursor.getCount()==0)return new ArrayList<>();  
  
 while (cursor.moveToNext()){  
 Task task = new Task(  
 cursor.getString(1),  
 cursor.getLong(2),  
 cursor.getString(3));  
  
 tasks.add(task);  
 }  
  
 return tasks;  
}

**Basic Blocks**

1. db = this.getWriteableDatabase();

tasks = new ArrayList<>();

cursor = db.rawQuery(“SELECT \* FROM “ + TABLE\_NAME, null);

1. if(cursor.getCount() == 0)
2. return new ArrayList<>();
3. while(cursor.moveToNext())
4. task = new Task(cursor.getString(1), cursor.getLong(2), cursor.getString(3);

tasks.add(task);

1. return tasks;

**Entry Node(s)**

1

**Exit Node(s)**

3, 6

**Control Flow**

1->2

2->3, 2->4

4->5, 4->6

5->4

1. getDescription(str)

public String getDescription(String name){  
 SQLiteDatabase db = this.getWritableDatabase();  
  
 Cursor totalCursor = db.rawQuery("SELECT " + *COLUMN\_DESC* + " FROM " + *TABLE\_NAME* + " WHERE NAME = ?", new String[]{name});  
 totalCursor.moveToFirst();  
  
 return totalCursor.getString(0);  
}

**Basic Blocks**

1. db = this.getWriteableDatabase();

totalCursor = db.rawQuery(“SELECT “ + COLUMN\_DESC + “ FROM “ + TABLE\_NAME + “ WHERE NAME = ?”, new String[]{name});

totalCursor.moveToFirst();

1. return totalCursor.getString(0);

**Entry Node(s)**

1

**Exit Node(s)**

2

**Control Flow**

1->2