

◆ STEP 3: Creating a Function: GetTotalAid

```
CREATE OR REPLACE FUNCTION GetTotalAid(p_student_id NUMBER)
```

```
RETURN NUMBER IS
```

```
total NUMBER;
```

```
BEGIN
```

```
SELECT NVL(SUM(approved_amount), 0)
```

```
INTO total
```

```
FROM Application
```

```
WHERE student_id = p_student_id AND status = 'Approved';
```

```
RETURN total;
```

```
EXCEPTION
```

```
WHEN OTHERS THEN
```

```
RETURN -1;
```

```
END;
```

🧠 What It Does:

- Accepts a **student ID** as input
- Returns the **total approved aid** for that student
- Uses NVL(..., 0) to return 0 if nothing found
- If any error occurs (invalid ID, etc.), it safely returns -1

✅ Why It Matters:

This function can be reused in reports, views, or queries to **quickly summarize aid per student** — a common MIS

```
CREATE TABLE Application_Log (  
  log_id          NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  action_performed VARCHAR2(50),  
  application_id   NUMBER,  
  performed_by     VARCHAR2(100),  
  action_time      TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);  
  
UPDATE Application  
SET status = 'Approved', approved_amount = 450000  
WHERE application_id = 101;  
  
INSERT INTO Application_Log (action_performed, application_id, performed_by)  
VALUES ('UPDATE', 101, 'admin_user');
```

Script Output x

Task completed in 0.493 seconds

PL/SQL procedure successfully completed.

Table APPLICATION_LOG created.

1 row updated.

1 row inserted.

requirement

