| ID | Test Scenario | SQL Query | Status P \| F |
|---|---|---|---|
| | | | |
| | SCHEMA TESTING | select statements | |
| 1 | Verify if User table in PWA database | SELECT * FROM pwa."User"; OR Show tables; | P |
| 2 | Veify User table name conventions | SELECT * FROM pwa."User"; OR Show tables; | P |
| 3 | Verify total number of columns in User table | Select count(*) AS No_Of_Coloumns from information_schema.columns where table_name = pwa."User"; | F |
| 4 | Verify column names in User table | Select coloumn_name from information_schema.columns where table_name = pwa."User"; | F |
| 5 | Verify if data types of columns in User table is as in User properties | Select coloumn_name, data_type from information_schema.columns where table_name = pwa."User"; | F |
| 6 | Verify column size of User table | Select coloumn_name, column_type from information_schema.columns where table_name = pwa."User"; | F |
| 7 | Verify if there is a NULL field in User table | Select coloumn_name, is_nullable from information_schema.columns where table_name = pwa."User"; | F |
| 8 | Verify column keys with specific constraints ( Primary Key...) in User table | Select coloumn_name, column_key from information_schema.columns where table_name = pwa."User"; | F |
| 9 | Verify if there is any duplicate column in table | Select column_name, COUNT(Column_Name) AS count FROM table_name = "User" GROUP BY profile id HAVING COUNT(profile id) > 1; | |
| | | | |
| | DATA MAPPING / CRUD Testing | all types of queries | |
| | | | |
| 1 | Verify if user added profile is reflecting in User table | INSERT INTO pwa."User" (id, "phoneNumber", "role", "isPhoneNumberVerified", "isNewUser", "isActive", "createdAt", "updatedAt", otp, "otpGeneratedAt", "userRole") VALUES(nextval('pwa."User_id_seq"'::regclass), '', 'JOB_SEEKER'::pwa."Role", false, true, true, CURRENT_TIMESTAMP, '', 0, '', 0); | |
| 2 | Verify if user deletes profile, it is deleted from User table as well | DELETE FROM pwa."User" WHERE id=nextval('pwa."User_id_seq"'::regclass); | |
| 3 | Verify if the user modifies specifc field in existing User, the change is reflected in User table | UPDATE pwa."User" SET "phoneNumber"='', "role"='JOB_SEEKER'::pwa."Role", "isPhoneNumberVerified"=false, "isNewUser"=true, "isActive"=true, "createdAt"=CURRENT_TIMESTAMP, "updatedAt"='', otp=0, "otpGeneratedAt"='', "userRole"=0 WHERE id=nextval('pwa."User_id_seq"'::regclass); | |
| 4 | Verify if user is able to retrieve all the changes made | SELECT id, "phoneNumber", "role", "isPhoneNumberVerified", "isNewUser", "isActive", "createdAt", "updatedAt", otp, "otpGeneratedAt", "userRole" FROM pwa."User"; | |
| | | | |
| | DATA INTEGRITY Testing | Insert queries | |
| | Positive cases | | |
| 1 | Verify Id | INSERT INTO pwa."User" (id, "phoneNumber", "role", "isPhoneNumberVerified", "isNewUser", "isActive", "createdAt", "updatedAt", otp, "otpGeneratedAt", "userRole") VALUES(nextval('pwa."User_id_seq"'::regclass), '', 'JOB_SEEKER'::pwa."Role", false, true, true, CURRENT_TIMESTAMP, '', 0, '', 0); | |
| 2 | Verify phone number | | |
| 3 | Verify role | | |
| 4 | Verify is phone number verified | | |
| 5 | Verify is New User | | |

| | | | |
|---|---|---|---|
| 6 | Verify is Active | | |
| 7 | Verify Created At | | |
| 8 | Verify Updated At | | |
| 9 | Verify OTP | | |
| 10 | Verify OTP Generated At | | |
| | | | |
| | Negative cases | | |
| 1 | Verify Id | INSERT INTO pwa."User"<br>(id, "phoneNumber", "role", "isPhoneNumberVerified", "isNewUser", "isActive", "createdAt", "updatedAt", otp, "otpGeneratedAt", "userRole")<br>VALUES(nextval('pwa."User_id_seq"'::regclass), '', 'JOB_SEEKER'::pwa."Role", false, true, true, CURRENT_TIMESTAMP, '', 0, '', 0); | |
| 2 | Verify phone number | | |
| 3 | Verify role | | |
| 4 | Verify is phone number verified | | |
| 5 | Verify is New User | | |
| 6 | Verify is Active | | |
| 7 | Verify Created At | | |
| 8 | Verify Updated At | | |
| 9 | Verify OTP | | |
| 10 | Verify OTP Generated At | | |
| | | | |
| | FUNCTIONAL TESTING | | |
| | | | |
| 1 | Verify if user can retrieve phone number based on user id | SELECT id, "phoneNumber", "role", "isPhoneNumberVerified", "isNewUser", "isActive", "createdAt", "updatedAt", otp, "otpGeneratedAt", "userRole" FROM pwa."User" where "User".id = 3; | P |
| 2 | Verify if user can retrieve role based on user id | | |
| 3 | Verify if user can retrieve details if phone number is verified based on user id | | |
| 4 | Verify if we can retrieve if the user is new based on user id | | |
| 5 | Verify if we can retrieve if the user is active based on user id | | |
| 6 | Verify if user can retrieve OTP based on user id | | |
| 7 | Verify if we can retrieve user created time based on user id | | |
| 8 | Verify if we can retrieve OTP generation time based on user id | | |
| | | | |