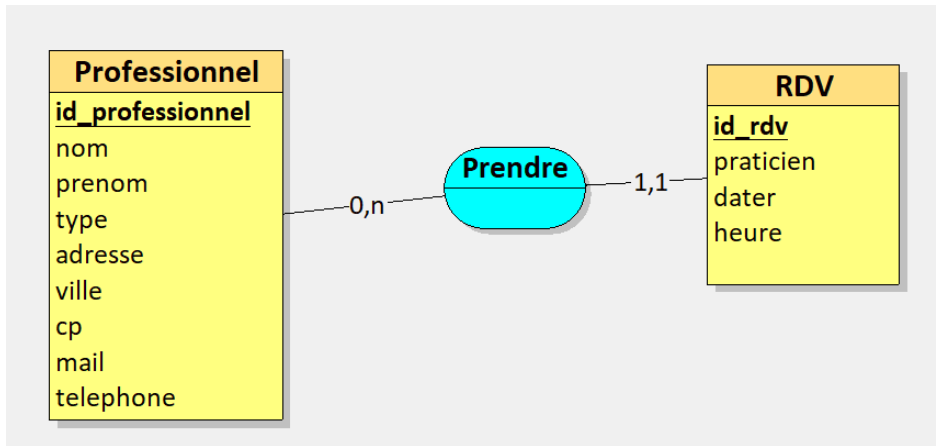


Le MDC :



Le MLD :

Professionnel (id_professionnel, nom, prenom, type, adresse, ville, cp, mail, telephone)

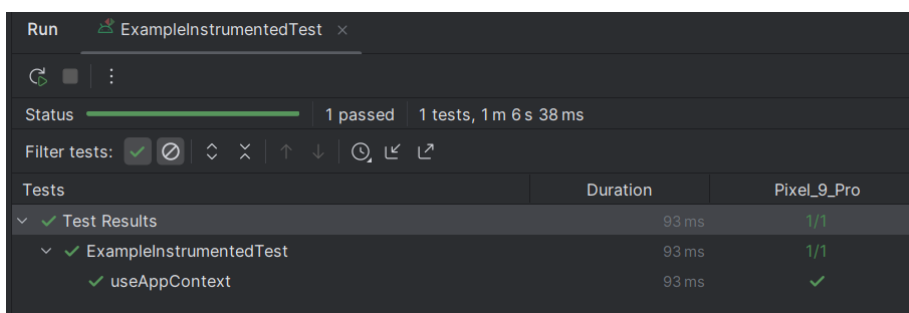
RDV (id_rdv, praticien, dater, heure, #id_professionnel)

Tests unitaires :

- Teste de l'insertion d'un professionnel dans la base de données :

```
private Base_donnee db;
@Test
public void useAppContext() {
    Context appContext = InstrumentationRegistry.getInstrumentation().getTargetContext();

    db = new Base_donnee(appContext);
    int nb1=0;
    Cursor c1 = db.getProfessionnel();
    nb1 = c1.getCount();
    c1.close();
    db.insertDataProfessionnel("TOTO", "Bob", "Médecin",
        "156 rue des fleures", "Toulouse", 31000, "bob.toto@mail.com", 601020304);
    Cursor c2 = db.getProfessionnel();
    int nb2 = c2.getCount();
    c2.close();
    assertEquals(nb1 + 1, nb2);
}
```



- Test de l'insertion d'un rendez-vous dans la base de données :

```

@Test
public void insertionrdv() {
    Context appContext = InstrumentationRegistry.getInstrumentation().getTargetContext();

    String spinnerPro = "PINCHOT Valentine (Médecin)";
    String curDate = "25/03/2026";
    String heureFormat = "14:30";

    db = new Base_donnee(appContext);
    int nb1=0;
    Cursor c1 = db.getRDV();
    nb1 = c1.getCount();
    c1.close();
    db.insertDatardv(spinnerPro, curDate, heureFormat);
    Cursor c2 = db.getRDV();
    int nb2 = c2.getCount();
    c2.close();
    assertEquals(nb1 + 1, nb2);
}

```

The screenshot shows the 'Run' tab in Android Studio for the test 'insertionrdv()'. The status bar indicates '1 passed' and '1 tests, 18 s 611 ms'. Below the status bar, there are filter icons and a table of test results.

Tests	Duration	Pixel_9_Pro
✓ Test Results	146 ms	1/1
✓ ExampleInstrumentedTest	146 ms	1/1
✓ insertionrdv	146 ms	✓

- Teste de rechercher un professionnel par un code postale :

```

@Test
public void testChercherProfessionnelParCP() {
    Context appContext = InstrumentationRegistry.getInstrumentation().getTargetContext();
    Base_donnee db = new Base_donnee(appContext);

    // On insère un professionnel "type" pour être sûr d'avoir une donnée connue
    // On utilise un CP unique pour le test, ex: 99999
    String nomTest = "Dupond";
    String villeTest = "Toulouse";
    int cpTest = 99999;

    db.insertDataProfessionnel(nomTest, "Jean", "Médecin",
        "10 rue des champs", villeTest, cpTest, "jean.dupond@mail.com", 102030405);

    // On appelle la méthode chercherProfessionnel avec le CP unique
}

```

```

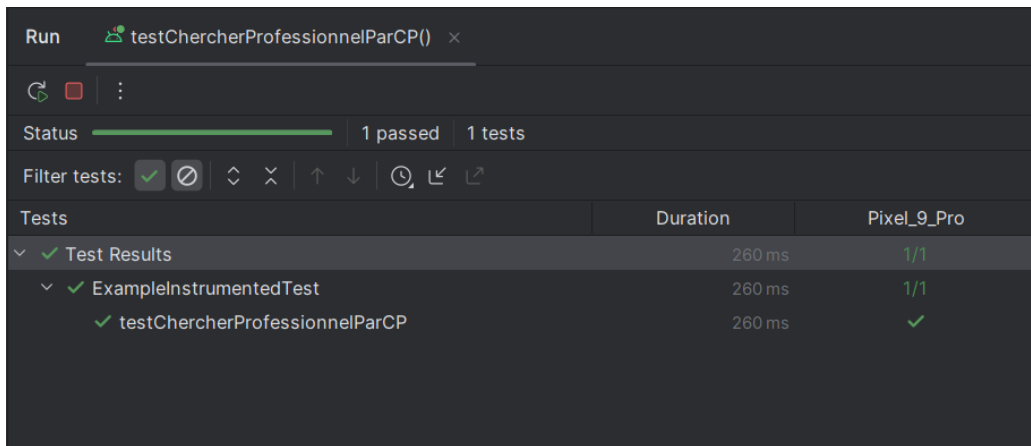
Cursor cursor = db.chercherProfessionnel(String.valueOf(cpTest));

// On vérifie que le curseur n'est pas null et qu'il y a au moins 1 résultat
assertNotNull("Le curseur ne doit pas être null", cursor);
assertTrue("On devrait trouver au moins le professionnel inséré", cursor.getCount() > 0);

// On vérifie que les données trouvées sont les bonnes
if (cursor.moveToFirst()) {
    String nomRecupere = cursor.getString(1);
    assertEquals("Le nom trouvé doit être TEST_NOM", nomTest, nomRecupere);
}

cursor.close();
db.close();
}

```



- [Teste de rechercher un rdv selon une date précise, s'il y a un rdv ou non](#)

@Test

```

public void testGetRdvParDateAvecJointure() {
    Context appContext = InstrumentationRegistry.getInstrumentation().getTargetContext();
    Base_donnee db = new Base_donnee(appContext);

    // On insère un professionnel "type"
    String nomPro = "VALENTIN";
    String villePro = "Toulouse";
    db.insertDataProfessionnel(nomPro, "Lucie", "Dentiste",
        "1 rue de la Paix", villePro, 31000, "lucie@mail.com", 601020304);

    // On insère un rendez-vous lié à ce professionnel
    String praticienRDV = "VALENTIN Lucie (Dentiste)";
    String dateRDV = "12/05/2026";
    String heureRDV = "14:00";
    db.insertDatardv(praticienRDV, dateRDV, heureRDV);

    Cursor cursor = db.getRdvParDate(dateRDV);

    assertNotNull("Le curseur ne doit pas être null", cursor);
}

```

```

assertTrue("On devrait trouver au moins 1 RDV pour cette date", cursor.getCount() > 0);

if (cursor.moveToFirst()) {
    // Vérification des données de la table RDV
    assertEquals("Le praticien doit correspondre", praticienRDV, cursor.getString(1));

    // Vérification des données jointes de la table Professionnel
    String adresseTrouvee = cursor.getString(8);
    String villeTrouvee = cursor.getString(9);

    assertEquals("L'adresse jointe doit être correcte", "1 rue de la Paix", adresseTrouvee);
    assertEquals("La ville jointe doit être correcte", villePro, villeTrouvee);
}
cursor.close();
db.close();
}

```

The screenshot shows the 'Run' window of an IDE. At the top, it says 'Run testGetRdvParDateAvecJointure()'. Below that, there's a status bar with a green progress indicator, '1 passed', and '1 tests, 32 s 565 ms'. A filter tests section shows a green checkmark and a magnifying glass icon. The main part of the window is a table of test results.

Tests	Duration	Pixel_9_Pro
✓ Test Results	257 ms	1/1
<ul style="list-style-type: none"> ✓ ExampleInstrumentedTest <ul style="list-style-type: none"> ✓ testGetRdvParDateAvecJointure 	257 ms	1/1