

Android Emulator Yerine Cihaz Kullanılırken Oluşan Veritabanını Bilgisayara Kopyalamak

Android cihazınızı **emulator** olarak kullanırken bazı problemler ortaya çıkmaktadır. Bunlardan bir tanesi veritabanı işlemlerinin yapıldığı uygulamalarda karşınıza çıkmaktadır. Veritabanı işlemleri yaparken genelde veritabanının **içeriğini** görmeye ihtiyaç duyarız. Bunun için ise veritabanı dosyasını bilgisayarımıza **indirmek** gereklidir. Eğer sizin android cihazınız **rooted** değilse **private** olan veritabanı dosyasını file manager'dan göremezsiniz. Bunun sonucunda ise bilgisayarınıza bu veritabanı dosyasını gönderemezsiniz.

Bu problemin çözümü için **Android SDK** içerisindeki **platform-tools** klasöründe bulunan **adb.exe** uygulamasına ihtiyacınız vardır. Cihazınız **rooted** olmadığı halde gizli veritabanını bilgisayarınıza **kopyalamak** için bazı komutların çalıştırılması gerekmektedir. **platform-tools** klasöründe **shift tuşu** ile birlikte **sağ** tıklarsanız **Open Command Window Here (Komut Penceresini Burada Aç)** seçeneği gelir. Bu seçeneğe tıklarsanız bulunduğunuz dizinde konsol ekranı açılır. Bu ekranda şu adımları sırasıyla yapınız:

```
1 | adb shell
2 | run-as paketAdı cat /data/data/paketAdı/databases/veritabanıAdı > /sdcard/veritabanıAdı.sqlite
```

Bu komutları sırasıyla uyguladığınızda **sdcard** içerisine veritabanı kopyalanmış olur. Bu işlemden sonra **Ctrl+C** tuşuna basarak önceki uygulamayı kapatınız. Daha sonra aşağıdaki komutu çalıştırınız:

```
1 | adb pull /sdcard/veritabanıAdı.sqlite C:\Users\{UserName}\Desktop\veritabanıAdı.sqlite
```

Tüm bu işlemlerden sonra artık Desktop'ta **veritabanıAdı.sqlite** dosyası kaydedilmiş olur.

Örnek bir uygulamada veritabanı adı ve paket adı şu şekilde olsun:

Veritabanı Adı: users

Paket Adı: mucayufa.android.database.simple

O zaman çalıştırılması gereken komutlar şu şekilde olacaktır:

```
1 | adb shell
2 | run-as mucayufa.android.database.simple cat /data/data/mucayufa.android.database.simple/databases/users > /sdcard/users.sqlite
```

Ctrl+C'ye basıldıktan sonra

```
1 | adb pull /sdcard/users.sqlite C:\Users\{Kullanici}\Desktop\users.sqlite
```

komutu çalıştırılırsa, Masaüstünde **users.sqlite** isimli bir dosya oluşmuş olur.

Not: Eğer yukarıdaki işlemlerde **package unknown hatası** alırsanız, iki farklı yöntemle veritabanını kopyalayabilirsiniz:

Yöntem 1: Cihazınız **rooted** ise aşağıdaki komutları sırasıyla çalıştırınız:

1. platform-tools klasöründe komut penceresi açılır

2. Aşağıdaki komutlar sırasıyla çalıştırılır

```
1 | adb shell
2 | su
3 | run-as packageName cat /data/data/packageName/databases/users > /sdcard/users.sqlite
```

Ctrl+C'ye basıldıktan sonra

```
1 | adb pull /sdcard/users.sqlite C:\Users\{Kullanici}\Desktop\users.sqlite
```

Not: Bu komutları bir **batch** dosyası (.bat ile biten dosya) oluşturup, bu dosyaya kopyaladıktan sonra **platform-tools** klasörü içerisinde çalıştırabilirsiniz:

```
1 | cls
2 | @ECHO OFF
3 | adb shell "su -c 'run-as paketAdi cat /data/data/paketAdi/databases/users > /sdcard/users.sqlite'"
```

```
4 | adb pull /sdcard/users.sqlite C:\Users\{Kullanici}\Desktop\users.sqlite
```

Yöntem 2: Kodlar ile veritabanını **import** veya **export** edebiliriz:

```
1 | import android.os.Environment;
2 | import android.widget.Toast;
3 |
4 | import java.io.File;
5 | import java.io.FileInputStream;
6 | import java.io.FileOutputStream;
7 | import java.nio.channels.FileChannel;
8 |
9 | public class ExportImportDB {
10 |     public ExportImportDB() {
11 |
12 |         //Backup'i alınacak veritabanı için BackupFolder isiminde bir klasör yaratılır.
13 |         File direct = new File(Environment.getExternalStorageDirectory() + "/BackupFolder");
14 |
15 |         if (!direct.exists()) {
16 |             if (direct.mkdir()) {
17 |                 //directory is created;
18 |             }
19 |
20 |         }
21 |     }
22 |
23 |     //importing database
24 |     private void importDB() {
25 |         try {
26 |             File sd = Environment.getExternalStorageDirectory();
27 |             File data = Environment.getDataDirectory();
28 |             String packageName="android uygulamasının paket adı. Örnek: mucayufa.android.database.simple";
29 |             String veritabaniAdi="android uygulamasının veritabanı adı. Örnek: users";
30 |             if (sd.canWrite()) {
31 |                 String currentDBPath = "//data//" + packageName
32 |                     + "//databases//" + veritabaniAdi;
33 |                 String backupDBPath = "/BackupFolder/"+veritabaniAdi+".sqlite";
34 |                 File backupDB = new File(data, currentDBPath);
35 |                 File currentDB = new File(sd, backupDBPath);
36 |
37 |                 FileChannel src = new FileInputStream(currentDB).getChannel();
38 |                 FileChannel dst = new FileOutputStream(backupDB).getChannel();
39 |                 dst.transferFrom(src, 0, src.size());
40 |                 src.close();
41 |                 dst.close();
42 |
43 |             }
44 |         } catch (Exception e) {}
45 |     }
46 |
47 |     //exporting database
48 |     private void exportDB() {
49 |         try {
50 |             File sd = Environment.getExternalStorageDirectory();
51 |             File data = Environment.getDataDirectory();
52 |             String packageName="android uygulamasının paket adı. Örnek: mucayufa.android.database.simple";
53 |             String veritabaniAdi="android uygulamasının veritabanı adı. Örnek: users";
54 |             if (sd.canWrite()) {
55 |                 String currentDBPath = "//data//" + packageName
56 |                     + "//databases//" + veritabaniAdi;
57 |                 String backupDBPath = "/BackupFolder/"+veritabaniAdi+".sqlite";
58 |                 File currentDB = new File(data, currentDBPath);
59 |                 File backupDB = new File(sd, backupDBPath);
60 |
61 |                 FileChannel src = new FileInputStream(currentDB).getChannel();
62 |                 FileChannel dst = new FileOutputStream(backupDB).getChannel();
63 |                 dst.transferFrom(src, 0, src.size());
64 |                 src.close();
65 |                 dst.close();
66 |                 Toast.makeText(null, backupDB.toString(),
67 |                     Toast.LENGTH_LONG).show();
68 |
69 |             }
70 |         } catch (Exception e) {}
71 |     }
72 | }
73 | }
```