



**Free Questions for Associate-Android-Developer by
ebraindumps**

Shared by Franco on 24-05-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

By default, the notification's text content is truncated to fit one line. If you want your notification to be longer, for example, to create a larger text area, you can do it in this way:

Options:

A- NotificationCompat.Builder builder = new NotificationCompat.Builder(this, CHANNEL_ID)
.setContentText('Much longer text that cannot fit one line...')
.setStyle(new NotificationCompat.BigTextStyle()
.bigText('Much longer text that cannot fit one line...'))

...

B- NotificationCompat.Builder builder = new NotificationCompat.Builder(this, CHANNEL_ID)
.setContentText('Much longer text that cannot fit one line...')
.setLongText('Much longer text that cannot fit one line...'))

...

C- NotificationCompat.Builder builder = new NotificationCompat.Builder(this, CHANNEL_ID)
.setContentText('Much longer text that cannot fit one line...')
.setTheme(android.R.style.Theme_LongText);

...

Answer:

A

Explanation:

<https://developer.android.com/training/notify-user/build-notification>

Question 2

Question Type: MultipleChoice

What is a correct part of an Implicit Intent for sharing data implementation?

Options:

A- Intent sendIntent = new Intent(this, UploadService.class) sendIntent.putExtra(Intent.EXTRA_TEXT, textMessage);

B- Intent sendIntent = new Intent(); sendIntent.setType(Intent.ACTION_SEND);

C- Intent sendIntent = new Intent(this, UploadService.class) sendIntent.setData(Uri.parse(fileUrl));

D- Intent sendIntent = new Intent(); sendIntent.setAction(Intent.ACTION_SEND);

Answer:

D

Explanation:

Create the text message with a string

```
Intent sendIntent = new Intent(); sendIntent.setAction(Intent.ACTION_SEND); sendIntent.putExtra(Intent.EXTRA_TEXT, textMessage);  
sendIntent.setType('text/plain');
```

<https://developer.android.com/guide/components/fundamentals>

Question 3

Question Type: MultipleChoice

Select four different types of app components. (Choose four.)

Options:

A- Application

B- Layouts

C- Activities

D- Services

E- AlarmManager

F- WorkManager

G- Broadcast receivers

H- Content providers

I- Fragments

Topic 2, Java Exam

Answer:

C, D, G, H

Question 4

Question Type: MultipleChoice

About queries in DAO classes. Room verifies the return value of the query such that if the name of the field in the returned object doesn't match the corresponding column names in the query response, Room alerts you in one of the following two ways: (Choose two.)

Options:

- A- It gives a warning if no field names match.
- B- It gives a warning if only some field names match.
- C- It gives an error if no field names match.
- D- It gives an error if only some field names match.

Answer:

B, C

Question 5

Question Type: MultipleChoice

@Query is the main annotation used in DAO classes. It allows you to perform read/write operations on a database. Each @Query method is verified at compile time, so what happens if there is a problem with the query?

Options:

- A- a runtime error occurs instead of a compilation failure.
- B- a compilation error occurs instead of a runtime failure.
- C- both compilation error and runtime failure occurs.

Answer:

B

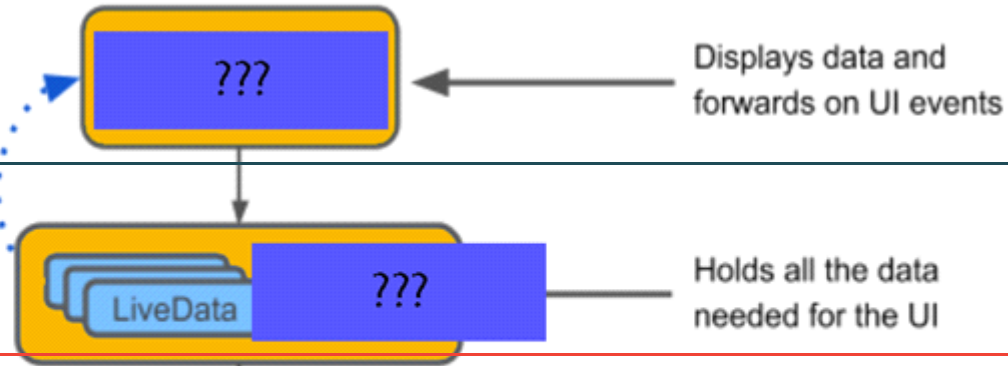
Question 6

Question Type: DragDrop

The diagram below shows a basic form of the recommended architecture for apps that use Architecture Components. The architecture consists of a UI controller, a ViewModel that serves LiveData, a Repository, and a Room database. Drag modules to correct places.

Answer:

UI is notified of changes using observation

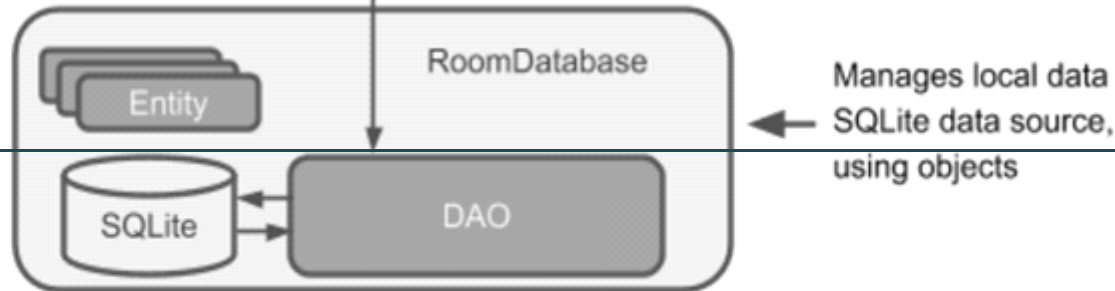


Displays data and forwards on UI events

Holds all the data needed for the UI

Single source of truth for all app data; clean API for UI to communicate with

A class that you create for managing multiple data sources. In addition to a Room database, this class could manage remote data sources such as a web server. It is about:



Manages local data SQLite data source, using objects

Options:

- A- Activity/Fragment
- B- ViewModel
- C- Repository
- D- Room database Repository



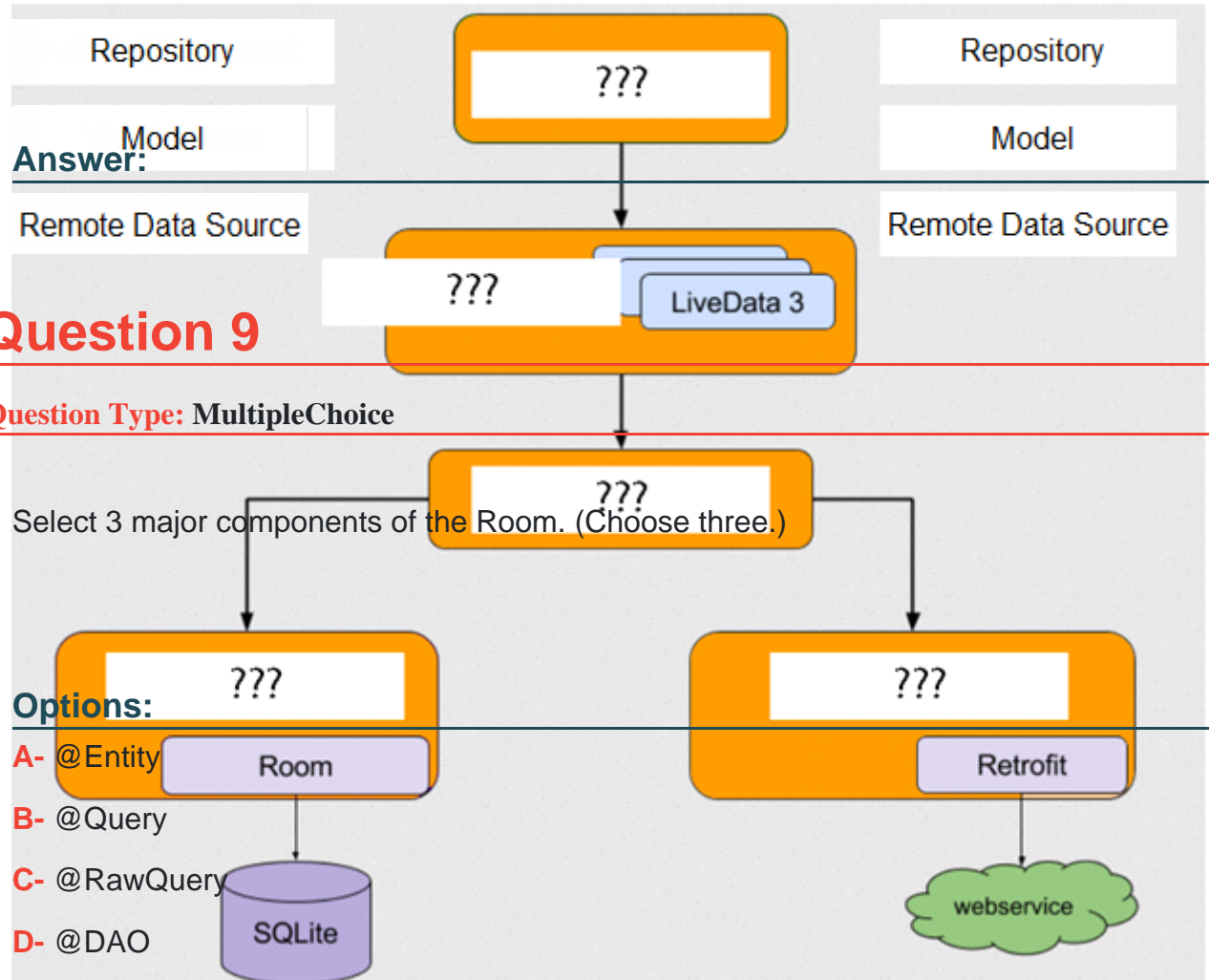
Answer:

C

Question 8

Question Type: DragDrop

With recommended app architecture. Fill the following diagram, which shows how all the modules usually should interact with one another after designing the app (drag modules to correct places).



Repository
Model
Answer:

Repository
Model

Remote Data Source

Remote Data Source

Question 9

Question Type: MultipleChoice

Select 3 major components of the Room. (Choose three.)

Options:

- A- @Entity
- B- @Query
- C- @RawQuery
- D- @DAO
- E- @WorkerThread
- F- @Database

Answer:

A, D, F

Question 10

Question Type: MultipleChoice

If you added to your build.gradle file a room.schemaLocation:

```
android {  
  
    defaultConfig {  
  
        javaCompileOptions {  
  
            annotationProcessorOptions {  
  
                arguments = ["room.schemaLocation": "$projectDir/schemas".toString  
  
                ()]  
  
            }  
  
        }  
  
    }  
  
}
```

```
}
```

Then, you build your app or module.

As a result you got a json file, with such path to it: `app/schemas/your_app_package/db_package/DbClass/DB_VERSION.json` What are the correct statements about this file? (Choose all that apply.)

Options:

- A-** It's a file with Room-exported schema
- B-** Main JSONObject in this file usually should contain a number 'formatVersion' and a JSONObject 'database'
- C-** The JSONObject 'database' in this file usually should contain such objects, like 'entities', 'views', 'setupQueries', etc.

Answer:

A, B, C

Explanation:

Exported schema file example:

```
{
```

```
'formatVersion': 1,
```

```
'database': {  
  
  'version': 1,  
  
  'identityHash': 'd90c93040756d2b94a178d5555555555',  
  
  'entities': [  
  
    {  
  
      'tableName': 'tea_table',  
  
      'createSql': 'CREATE TABLE IF NOT EXISTS `${TABLE_NAME}` (`id` INTEGER  
PRIMARY KEY AUTOINCREMENT NOT NULL, `name` TEXT, `type` TEXT, `origin` TEXT,  
`steep_times` INTEGER, `Description` TEXT, `ingredients` TEXT, `cafeinLevel` TEXT, `favorite` INTEGER)',  
  
      'fields': [  
  
        {  
  
          'fieldPath': 'mId',  
  
          'columnName': 'id',  
  
          'affinity': 'INTEGER',  
  
          'notNull': true
```

```
},  
  
{  
  'fieldPath': 'mName',  
  'columnName': 'name',  
  'affinity': 'TEXT',  
  'notNull': false  
},  
  
{  
  'fieldPath': 'mType',  
  'columnName': 'type',  
  'affinity': 'TEXT',  
  'notNull': false  
},  
  
{  
  'fieldPath': 'mOrigin',
```

```
'columnName': 'origin',  
  
'affinity': 'TEXT',  
  
'notNull': false  
  
},  
  
{  
  
'fieldPath': 'mSteepTimeMs',  
  
'columnName': 'steep_times',  
  
'affinity': 'INTEGER',  
  
'notNull': false  
  
},  
  
{  
  
'fieldPath': 'mDescription',  
  
'columnName': 'Description',  
  
'affinity': 'TEXT',  
  
'notNull': false
```

```
},  
  
{  
  'fieldPath': 'mIngredients',  
  'columnName': 'ingredients',  
  'affinity': 'TEXT',  
  'notNull': false
```

```
},  
  
{  
  'fieldPath': 'mCaffeineLevel',  
  'columnName': 'cafeinLevel',  
  'affinity': 'TEXT',  
  'notNull': false
```

```
},  
  
{  
  'fieldPath': 'mFavorite',
```



```
'columnName': 'favorite',
```

```
'affinity': 'INTEGER',
```

```
'notNull': false
```

```
}
```

```
],
```

```
'primaryKey': {
```

```
'columnNames': [
```

```
'id'
```

```
],
```

```
'autoGenerate': true
```

```
},
```

```
'indices': [],
```

```
'foreignKeys': []
```

```
}
```

```
],
```

```
'views': [],  
  
'setupQueries': [  
  
'CREATE TABLE IF NOT EXISTS room_master_table (id INTEGER PRIMARY  
KEY,identity_hash TEXT)',  
  
'INSERT OR REPLACE INTO room_master_table (id,identity_hash) VALUES(42,  
'd90c93040756d2b94a178d5555555555)'  
  
]  
  
}  
  
}
```

To Get Premium Files for Associate-Android-Developer Visit

<https://www.p2pexams.com/products/associate-android-developer>

For More Free Questions Visit

<https://www.p2pexams.com/google/pdf/associate-android-developer>

