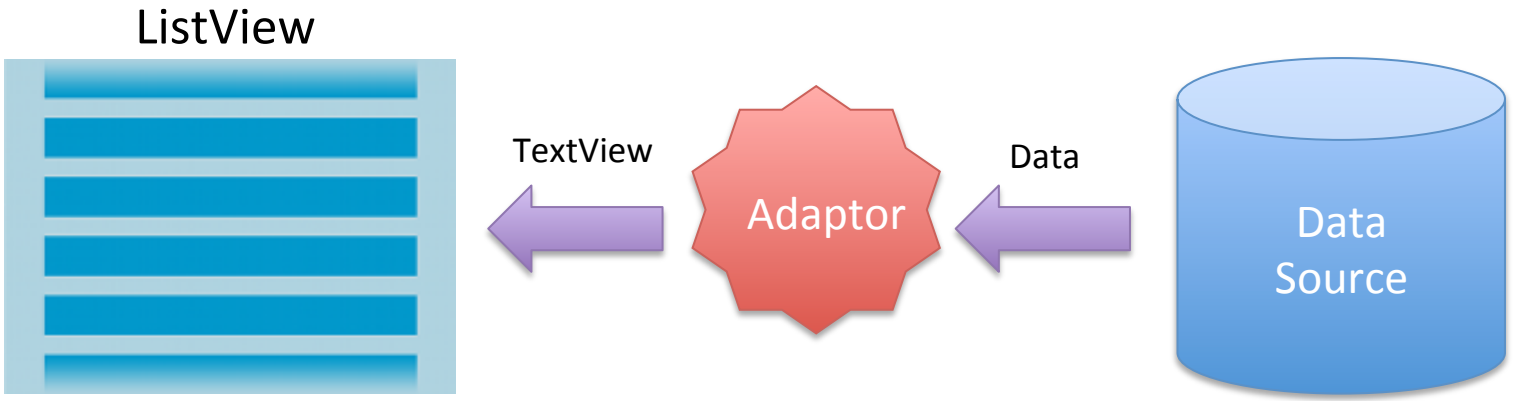
# Storage/ SQLite

# Layouts with Dynamic Data

- use AdaptorView when data is dynamic
  - Layout is dynamically determined by the data
  - Adaptor fills the AdaptorView with data
  - AdaptorView subclasses: ListView, GridView
  - Data: array, cursor from content providers, DB
- Types
  - ArrayAdaptor: from an array
  - SimpleCursorAdaptor: from a cursor



# ArrayAdaptor

- For each item of an array, call toString(), insert a TextView
- Usage
  - Create an Adaptor
    - new ArrayAdaptor<T>(<context>, <layout>, <array>)
      - <layout>: layout for each row, containing a TextView
    - Android's default row layouts: R.simple_list_*
      - http://developer.android.com/reference/android/R.layout.html
      - https://github.com/android/platform_frameworks_base/tree/master/core/res/res/layout
      - <SDK>/platforms/android-*/data/res/layout/
  - Register the adaptor
    - <list view>.setAdaptor(<adaptor>)
  - When data changed
    - <adaptor>.notifyDataSetChanged();
- Customization
  - Look: Define your own row layout
  - Data representation: Override toString() of the element
  - Different View type (e.g., ImageView): extend ArrayAdaptor and override getView()
- Example
  - ArrayAdapter<String> ad = new ArrayAdapter<String>(this, R.layout.simple_list_item_1, myArray);
    ((ListView) findViewById(R.id.listview)).setAdapter(ad);

# SimpleCursorAdaptor

- Data from a cursor (from ContentProvider, SQLite, ..)
- Bind each column to each view in row layout
  - Define a string array of column names
    - String[] fromColumns = { <col1>, .. }
  - Define an integer array of corresponding view ids
    - Int[] toViews = {R.id.<id1>, .. }
- Create an adaptor
  - new SimpleCursorAdapter(this, <row layout>, <cursor>, <col-array>, <viewID-array>, <flags>);
- Register the adaptor
  - <list view>.setAdaptor(<adaptor>)
- Handling click events
  - <list view>.setOnItemClickListener( new OnItemClickListener() {
    public void onItemClick(AdapterView parent, View v, int position, long id{..}
    } );

# Matrix Cursor

- What if a static data with multiple columns?
  - Use matrix cursor on array, instead of DB
- How to
  - Create a column-name array
    - String[] <cols>= { “<col1>”, .. }
  - Create a matrix cursor
    - MatrixCursor <mc>= new MatrixCursor(<cols>);
    - startManagingCursor(<mc>);
  - Add data
    - <mc>.addRow(new Object[] { <data1>, .. });
  - Create SimpleCursorAdaptor and register it
    - new SimpleCursorAdapter( <context>, <row layout>, <mc>, <cols>, <viewID-array>);

# SQLite

- Create a database (or open if exists)
  - *SQLiteDatabase db = openOrCreateDatabase( <db_name>, <mode>, <cursor> )*
- Run SQL
  - *db.execSQL(<sql>)*
  - *http://www.w3schools.com/sql/*
- Run SQL with cursor
  - *Cursor <cur> = db.rawQuery( <SQL>, null )*
  - *<cur>.getString( <cur>.getColumnIndex( <col-name> ) )*
  - *do {..} while( <cur>.moveToNext() );*
- Create a table
  - *create table if not exists <table-name> (<colname> <type>, .. )*
- Insert values
  - *insert into <table> values (<col value>,…, <col value>)*
- Get values
  - select * from <table>
- Close the database
  - *db.close*
- Stored in
  - /data/data/<pkg>/databases/<db-name>

# Database Example

- students table

| _id (Int) | name (Text) | age (Int) | dept (Text) |
|---|---|---|---|
| 1 | Obama | 25 | CE |
| 2 | Trump | 22 | IC |

Data Types
- NULL
- INT
- REAL
- TEXT
- BLOB

```
CREATE TABLE IF NOT EXISTS STUDENTS (
    _id         INT PRIMARY KEY     NOT NULL,
    name        TEXT    NOT NULL,
    age         INT     NOT NULL,
    dept        TEXT
);
```

# Caution

- If use SimpleCursorAdaptor
  - Add “_id” integer column in the table
  - Include “_id” in select statement