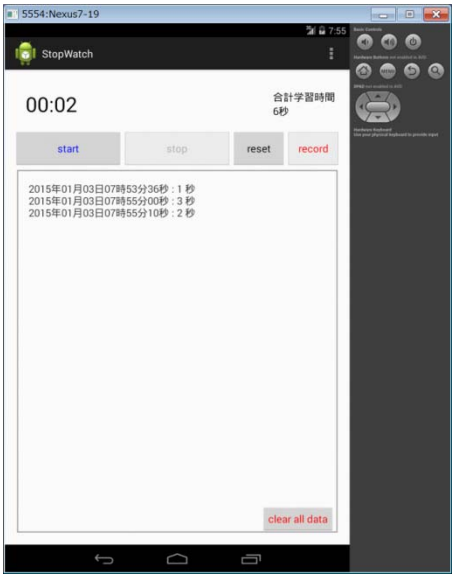
activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="kindai.so.stopwatch.MainActivity" >

    <Chronometer
        android:id="@+id/chronometer1"
        android:layout_alignParentLeft="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:textSize="36sp"
        android:text="Chronometer" />

    <LinearLayout
        android:id="@+id/linearLayout2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/chronometer1"
        android:layout_alignParentRight="true"
        android:layout_marginRight="10dp"
        android:orientation="vertical" >

    <TextView
      android:id="@+id/totalLabel"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignRight="@+id/linearLayout2"
      android:text="合計学習時間"
      android:textAppearance="?android:attr/textAppearanceMedium" />

     <TextView
      android:id="@+id/totalTV"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignRight="@+id/linearLayout2"
      android:text=""
      android:textAppearance="?android:attr/textAppearanceMedium" />

  </LinearLayout>

  <LinearLayout
```



```
    android:id="@+id/linearLayout1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/chronometer1"
    android:orientation="horizontal">

  <Button
      android:id="@+id/startButton"
      android:layout_width="0dp"
      android:layout_height="wrap_content"
      android:layout_weight="2"
      android:paddingTop="20dp"
      android:paddingBottom="20dp"
      android:textColor="#0000ff"
      android:text="start" />

  <Button
      android:id="@+id/stopButton"
      android:layout_width="0dp"
      android:layout_height="fill_parent"
      android:layout_weight="2"
      android:enabled="false"
      android:text="stop" />

   <Button
      android:id="@+id/resetButton"
      android:layout_width="0dp"
      android:layout_height="fill_parent"
      android:layout_weight="1"
       android:padding="0dp"
       android:enabled="false"
      android:text="reset" />

   <Button
      android:id="@+id/recordButton"
      android:layout_width="0dp"
      android:layout_height="fill_parent"
      android:layout_weight="1"
       android:padding="0dp"
       android:enabled="false"
       android:textColor="#ff0000"
      android:text="record" />
</LinearLayout>

  <TextView
    android:id="@+id/recordTV"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_below="@+id/linearLayout1"
    android:layout_centerHorizontal="true"
```

```
        android:layout_marginTop="10dp"
        android:padding="20dp"
        android:textSize="18sp"
        android:background="@drawable/border2dp"
        android:layout_margin="5dp"
        android:scrollbars="vertical"
        android:scrollbarAlwaysDrawVerticalTrack="true"
        />

    <Button
        android:id="@+id/clearAllButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="5dp"
        android:layout_alignRight="@+id/recordTV"
        android:layout_marginRight="5dp"
        android:text="clear all data"
        android:textColor="@color/red" />

</RelativeLayout>
```

MainActivity.java ----------

```
package kindai.so.stopwatch;

import java.text.SimpleDateFormat;
import java.util.Date;

import android.app.Activity;
import android.os.Bundle;
import android.os.SystemClock;
import android.text.method.ScrollingMovementMethod;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Chronometer;
import android.widget.TextView;

import android.content.ContentValues;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;

public class MainActivity extends Activity {
        private Chronometer chronometer;
        private Button startButton;
        private Button stopButton;
        private Button resetButton;
        private Button recordButton;
```

```
        private Button clearAllButton;
//時刻は  SystemClock.elapsedRealtime()  で起動してからの経過時間で取得
        private long startTime;  //startButtonを押した時刻
        private long stopTime;   //stopButtonを押した時刻
        private long countTime;  //stopWatchで計測した時間
        private long totalStudyTime;  //総学習時間

        private TextView recordTV;     //学習時間の記録を表示するTextView
        private TextView totalTV;      //総学習時間を表示するTextView

        private SQLiteDatabase studyDB;
        private String studyDataTableName;
        private StudeyDBOpenHelper helper;

        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_main);
                        chronometer  = (Chronometer)findViewById(R.id.chronometer1);
                        startButton  = (Button)findViewById(R.id.startButton);
                        stopButton   = (Button)findViewById(R.id.stopButton);
                        resetButton  = (Button)findViewById(R.id.resetButton);
                        recordButton = (Button)findViewById(R.id.recordButton);
                        clearAllButton = (Button)findViewById(R.id.clearAllButton);
                        recordTV = (TextView)findViewById(R.id.recordTV);

        recordTV.setMovementMethod(ScrollingMovementMethod.getInstance());
                        totalTV =  (TextView)findViewById(R.id.totalTV);

                //データベース読み込み表示
                helper = new StudeyDBOpenHelper(this);
                        studyDB = helper.getReadableDatabase();
                        studyDataTableName = helper.getStudyDataTableName();
                getStudyDataAndDisplay();      //学習履歴を検索し表示，総学習時間も表示
                //studyDB.close();

//--- ボタンにイベント登録 ----------------------------------------------------
        recordButton.setOnClickListener(new OnClickListener(){
                public void onClick(View v){
                        recordAndReset( countTime );//学習時間を記録し，学習履歴を検索し表示
                        startTime = 0;  stopTime = 0;   countTime = 0;
                                        startButton.setEnabled(true);
                                stopButton.setEnabled(false);
                                resetButton.setEnabled(true);
                                recordButton.setEnabled(false);
                }
        });

        startButton.setOnClickListener(new OnClickListener(){
                        public void onClick(View v){
```

```
                        long baseTime = SystemClock.elapsedRealtime();
                        startTime = baseTime;
                        baseTime -= countTime; // countTimeだけ遅らせる
                        //countTimeだけ前の時刻を計測起点に設定
                        chronometer.setBase(baseTime);
                        chronometer.start();
                                startButton.setEnabled(false);
                                stopButton.setEnabled(true);
                                resetButton.setEnabled(true);
                                recordButton.setEnabled(false);
                        }
                });

                stopButton.setOnClickListener(new OnClickListener(){
                        public void onClick(View v){
                                chronometer.stop();
                                stopTime = SystemClock.elapsedRealtime();
                                countTime += (stopTime - startTime);
                                startButton.setEnabled(true);
                                stopButton.setEnabled(false);
                                resetButton.setEnabled(true);
                                recordButton.setEnabled(true);
                        }
                });

                resetButton.setOnClickListener(new OnClickListener(){
                        public void onClick(View v){
                                chronometer.stop();
                                chronometer.setBase(SystemClock.elapsedRealtime());
                                startTime = 0;  stopTime = 0;   countTime = 0;
                                startButton.setEnabled(true);
                                stopButton.setEnabled(false);
                                resetButton.setEnabled(false);
                                recordButton.setEnabled(false);
                        }
                });

                clearAllButton.setOnClickListener(new OnClickListener(){
                        public void onClick(View v){
                        studyDB.execSQL("DROP TABLE IF EXISTS " + studyDataTableName );
                                helper.onCreate(studyDB);
                                recordTV.setText("");
                        }
                });
        }  //----End-of-onCreate---------------------------------------------
//-- データベースに学習時間データ t を追加してリセット---------------
        private void recordAndReset(long t){
                t = t/1000;     //msからsに変換
                Date date = new Date();         //現在の時刻を取得
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy'年'MM'月'dd'日'kk'時'mm'分'ss'秒'");
```

```
            String dDB = sdf.format(date); //DB  date  の値
            long tDB = t;  //DB time の値  （単位：秒）  分にするなら  /60.0
            insertDB(dDB, tDB);
            startTime = 0;  stopTime = 0;  countTime = 0;
        }

        public void insertDB(String d, long t){
            ContentValues values = new ContentValues();
                    values.put("date", d); values.put("time", t);
            studyDB.insertOrThrow( studyDataTableName, null, values);
        //String insertSql = "insert into " + studyDataTableName + " (date,time) values ";
        //studyDB.execSQL( insertSql + "('"+ d + "'," + t + ");"  );
            getStudyDataAndDisplay();
        }

        private void getStudyDataAndDisplay(){
          //テーブルから全データ抽出
           Cursor c = studyDB.query( studyDataTableName, new String[] { "date", "time" },
                                     null, null, null, null, null);
           boolean isNext = c.moveToFirst(); totalStudyTime = 0; int i = 0;
           while (isNext) {
                totalStudyTime += c.getInt(1);
                String s = String.format("%s : %d 秒", c.getString(0), c.getInt(1)) + "¥n";
                if( i== 0 ){ recordTV.setText( s ); }else{ recordTV.append( s ); }
               isNext = c.moveToNext(); i++;
           }
                c.close();    totalTV.setText( totalStudyTime + "秒" );
         }

        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
            // Inflate the menu; this adds items to the action bar if it is present.
            getMenuInflater().inflate(R.menu.main, menu);
            return true;
        }

        @Override
        public boolean onOptionsItemSelected(MenuItem item) {
            // Handle action bar item clicks here. The action bar will
            // automatically handle clicks on the Home/Up button, so long
            // as you specify a parent activity in AndroidManifest.xml.
            int id = item.getItemId();
            if (id == R.id.action_settings) {
                    return true;
            }
            return super.onOptionsItemSelected(item);
        }
}
```

StudeyDBOpenHelper.java----------

```
package kindai.so.stopwatch;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteDatabase.CursorFactory;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class StudeyDBOpenHelper extends SQLiteOpenHelper{
	final static private int DB_VERSION = 12;
	final private static String dbFileName ="study.db";
	final static String studyDataTableName = "studyData_table";
	String insertSql;

public StudeyDBOpenHelper(Context context) {
	// TODO Auto-generated constructor stub
	super(context, dbFileName, null, DB_VERSION);
}

@Override
public void onCreate(SQLiteDatabase db) {
	// TODO Auto-generated method stub
		String CREATE_TABLE = "create table " + studyDataTableName;
		CREATE_TABLE += " ( _id integer primary key autoincrement";
		CREATE_TABLE +=	",date text not null";
		CREATE_TABLE += ",time integer not null";
		CREATE_TABLE +=	");";
		insertSql = "insert into " + studyDataTableName + " (date,time) values";
		// table create
		db.execSQL( CREATE_TABLE );
		//  初期データ入力 table row insert
		//db.execSQL( insertSql + " ('2015 年 1 月 1 日 1 時 0 分 0 秒', 30);" );
	}

/*	public void insertData(String d, String t){
		myDB.execSQL( insertSql + " ('"+ d +"'," + t + ");"    );
	}
*/
	@Override
	public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
		// TODO Auto-generated method stub
	    db.execSQL("DROP TABLE IF EXISTS " + studyDataTableName );
	    //Log.d("db", "onUpgrade");
	    onCreate(db);
	}

	public String getStudyDataTableName(){ return studyDataTableName; }
}
```

SQLiteDatabase の明示的な close()を行わない理由は、SQLiteOpenHelper にある。SQLiteOpenHelper は、内部に SQLiteDatabase オブジェクトを保存しており、getWritableDatabase()で返す SQLiteDatabase オブジェクトは毎回同一のものとなる。つまり、SQLiteDatabase の close()を実行してしまうと、次回以降に同じ SQLiteOpenHelper を使ってデータベースを開いたときに、既に close()された状態のオブジェクトが返されてしまう。そのような事情から、アプリケーション終了時などの明白なタイミング以外は、SQLiteDatabase を close()しないように実装する場合が多い。