

## 4.0 T-SQL STATEMENTS

### 4.1 SCRIPTS FOR CREATION OF DATABASE

```
--CREATING DATABASE
USE master;
GO
CREATE DATABASE YoungLiveProject;

--CHOOSE THE DATABASE AS THE DATABASE IN USE
USE YoungLiveProject;
GO
```

### 4.2 SCRIPTS FOR CREATION OF TABLES

```
----CREATE TABLE FOR INDIA SCHEMA
CREATE TABLE [dbo].[india](
    [childid] [nvarchar](255) NOT NULL,
    [yc] [float] NULL,
    [round] [float] NOT NULL,
    [inround] [float] NULL,
    [panel] [float] NULL,
    [deceased] [nvarchar](255) NULL,
    [dint] [datetime] NULL,
    [clustid] [float] NULL,
    [commid] [nvarchar](255) NULL,
    [region] [float] NULL,
    [typesite] [float] NULL,
    [childloc] [float] NULL,
    [chsex] [float] NULL,
    [chlang] [float] NULL,
    [chethnic] [float] NULL,
    [chldrel] [float] NULL,
    [agemon] [float] NULL,
    [marrcohab] [nvarchar](255) NULL,
    [marrcohab_age] [nvarchar](255) NULL,
    [birth] [nvarchar](255) NULL,
    [birth_age] [nvarchar](255) NULL,
```

[chweight] [float] NULL,  
[chheight] [float] NULL,  
[bmi] [float] NULL,  
[underweight] [float] NULL,  
[stunting] [float] NULL,  
[thinness] [float] NULL,  
[bwght] [nvarchar](255) NULL,  
[bwdoc] [nvarchar](255) NULL,  
[numante] [nvarchar](255) NULL,  
[delivery] [nvarchar](255) NULL,  
[tetanus] [nvarchar](255) NULL,  
[bcg] [nvarchar](255) NULL,  
[measles] [nvarchar](255) NULL,  
[polio] [nvarchar](255) NULL,  
[dpt] [nvarchar](255) NULL,  
[hib] [nvarchar](255) NULL,  
[chmightdie] [nvarchar](255) NULL,  
[chillness] [nvarchar](255) NULL,  
[chinjury] [float] NULL,  
[chhprob] [float] NULL,  
[chdisability] [nvarchar](255) NULL,  
[chdisscale] [nvarchar](255) NULL,  
[chrephealth1] [nvarchar](255) NULL,  
[chrephealth2] [nvarchar](255) NULL,  
[chrephealth3] [nvarchar](255) NULL,  
[chrephealth4] [nvarchar](255) NULL,  
[chhrel] [float] NULL,  
[chhealth] [nvarchar](255) NULL,  
[cladder] [nvarchar](255) NULL,  
[preprim] [float] NULL,  
[agegr1] [float] NULL,  
[enrol] [float] NULL,  
[engrade] [float] NULL,  
[entype] [float] NULL,  
[hghgrade] [float] NULL,  
[timesch] [float] NULL,  
[levlread] [nvarchar](255) NULL,  
[levlwrit] [nvarchar](255) NULL,  
[literate] [nvarchar](255) NULL,

```

[careid] [float] NULL,
[caredu] [float] NULL,
[carehead] [float] NULL,
[careage] [float] NULL,
[caresex] [float] NULL,
[carerel] [float] NULL,
[carecantread] [nvarchar](255) NULL,
[careladder] [float] NULL,
[careldr4yrs] [float] NULL,
[dadid] [float] NULL,
[dadedu] [float] NULL,
[dadlive] [float] NULL,
[dadage] [float] NULL,
[dadcantread] [nvarchar](255) NULL,
[dadyrdied] [nvarchar](255) NULL,
[momid] [float] NULL,
[momedu] [float] NULL,
[momlive] [float] NULL,
[momage] [float] NULL,
[momcantread] [nvarchar](255) NULL,
[momyrdied] [nvarchar](255) NULL,
[headid] [float] NULL,
[headedu] [float] NULL,
[headage] [float] NULL,
[headsex] [float] NULL,
[headrel] [float] NULL,
[hhsiz] [float] NULL,
[wi] [float] NULL,
[hq] [float] NULL,
[sv] [float] NULL,
[cd] [float] NULL,
[drwaterq] [float] NULL,
[toiletq] [float] NULL,
[elecq] [float] NULL,
[cookingq] [float] NULL,
[ownhouse] [float] NULL,
[credit] [nvarchar](255) NULL,
[foodsec] [nvarchar](255) NULL,
CONSTRAINT [PK_indiaDF] PRIMARY KEY CLUSTERED

```

```
(
    [childid] ASC,
    [round] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
```

### 4.3 SCRIPTS FOR INSERTING DATA INTO TABLES

-----INSERT DATA INTO INDIA TABLE

```
INSERT INTO [dbo].[india](
    [childid],
    [yc],
    [round],
    [inround],
    [panel],
    [deceased],
    [dint],
    [clustid],
    [commid],
    [region],
    [typesite],
    [childloc],
    [chsex],
    [chlang],
    [chethnic],
    [chldrel],
    [agemon],
    [marrcohab],
    [marrcohab_age],
    [birth],
    [birth_age],
    [chweight],
    [chheight],
    [bmi],
    [underweight],
    [stunting],
```

[thinness],  
[bwght],  
[bwdoc],  
[numante],  
[delivery],  
[tetanus],  
[bcg],  
[measles],  
[polio],  
[dpt],  
[hib],  
[chmightdie],  
[chillness],  
[chinjury],  
[chhprob],  
[chdisability],  
[chdis scale],  
[chrephealth1],  
[chrephealth2],  
[chrephealth3],  
[chrephealth4],  
[chhrel],  
[chhealth],  
[cladder],  
[preprim],  
[agegr1],  
[enrol],  
[engrade],  
[entype],  
[hghgrade],  
[timesch],  
[levlread],  
[levlwrit],  
[literate],  
[careid],  
[caredu],  
[carehead],  
[careage],  
[caresex],

```

    [carere1],
    [carecantread],
    [careladder],
    [careldr4yrs],
    [dadid],
    [dadedu],
    [dadlive],
    [dadage],
    [dadcantread],
    [dadyrdied],
    [momid],
    [momedu],
    [momlive],
    [momage],
    [momcantread],
    [momyrdied],
    [headid],
    [headedu],
    [headage],
    [headsex],
    [headrel],
    [hhsizel],
    [wi],
    [hq],
    [sv],
    [cd],
    [drwaterq],
    [toiletq],
    [elecq],
    [cookingq],
    [ownhouse],
    [credit],
    [foodsec]
)
SELECT
[childid],
    [yc],
    [round],
    [inround],

```

[panel],  
[deceased],  
[dint],  
[clustid],  
[commid],  
[region],  
[typesite],  
[childloc],  
[chsex],  
[chlang],  
[chethnic],  
[chldrel],  
[agemon],  
[marrcohab],  
[marrcohab\_age],  
[birth],  
[birth\_age],  
[chweight],  
[chheight],  
[bmi],  
[underweight],  
[stunting],  
[thinness],  
[bwght],  
[bwdoc],  
[numante],  
[delivery],  
[tetanus],  
[bcg],  
[measles],  
[polio],  
[dpt],  
[hib],  
[chmightdie],  
[chillness],  
[chinjury],  
[chhprob],  
[chdisability],  
[chdissscale],

[chrephealth1],  
[chrephealth2],  
[chrephealth3],  
[chrephealth4],  
[chhrel],  
[chhealth],  
[cladder],  
[preprim],  
[agegr1],  
[enrol],  
[engrade],  
[entype],  
[hghgrade],  
[timesch],  
[levlread],  
[levlwrit],  
[literate],  
[careid],  
[caredu],  
[carehead],  
[careage],  
[caresex],  
[carerel],  
[carecantread],  
[careladder],  
[careldr4yrs],  
[dadid],  
[dadedu],  
[dadlive],  
[dadage],  
[dadcantread],  
[dadyrdied],  
[momid],  
[momedu],  
[momlive],  
[momage],  
[momcantread],  
[momyrdied],  
[headid],



```

    [headedu],
    [headage],
    [headsex],
    [headrel],
    [hhsize],
    [wi],
    [hq],
    [sv],
    [cd],
    [drwaterq],
    [toiletq],
    [elecq],
    [cookingq],
    [ownhouse],
    [credit],
    [foodsec]
FROM indiaDF

```

#### 4.4 CREATING VIEWS

The data table above contains a lot of numerical values which have stipulated meaning in the data dictionary so to make the data more meaningful, views comprising of a columns needed for this report was created with all the data definitions for all four countries. These four views will be joined together to produce one large view that is composed of information from all four countries. Paginated reports will subsequently be generated by querying these views. The script for creating this view for Ethiopia is as follow:

```

--VIEW THAT CONTAIN DATA DEFINITION FOR SOME FIELD IN THE ETHIOPIA
TABLE
CREATE VIEW ethiopia_data
AS
SELECT
CASE WHEN childid LIKE 'ET%' THEN 'ethiopia'
END AS country,
[childid],
CASE WHEN yc = 1 THEN 'younger'

```

```

ELSE 'older'
END AS [yc],
      [round],
      [clustid],
      [commid],
CASE WHEN region = 1 THEN 'Tigray'
WHEN region = 2 THEN 'Afar'
WHEN region = 3 THEN 'Amhara'
WHEN region = 4 THEN 'Oromiya'
WHEN region = 5 THEN 'Somali'
WHEN region = 6 THEN 'Benshangul'
WHEN region = 7 THEN 'SNNP'
WHEN region = 12 THEN 'Gambela'
WHEN region = 13 THEN 'Harari'
WHEN region = 14 THEN 'Addis Ababa'
WHEN region = 15 THEN 'Dire'
ELSE 'NULL'
END AS region,
CASE WHEN typesite = 1 THEN 'urban'
WHEN typesite = 2 THEN 'rural'
ELSE 'NULL'
END AS typesite,
CASE WHEN chsex = 1 THEN 'male'
WHEN chsex = 2 THEN 'female'
ELSE 'NULL'
END AS chsex,
      [agemon],
      [chweight],
      [chheight],
      [bmi],
CASE WHEN underweight = 0 THEN 'underweight'
WHEN underweight = 1 THEN 'moderately underweight'
WHEN underweight = 2 THEN 'severely underweight'
ELSE 'NULL'
END AS underweight,
CASE WHEN stunting = 0 THEN 'not stunted'
WHEN stunting = 1 THEN 'moderately stunted'
WHEN stunting = 2 THEN 'severely stunted'
ELSE 'NULL'

```

```

END AS stunting,
CASE WHEN thinness = 0 THEN 'not thin'
WHEN thinness = 1 THEN 'moderately thin'
WHEN thinness = 2 THEN 'severely thin'
ELSE 'NULL'
END AS thinness,
    [bwght],
CASE WHEN delivery = 0 THEN 'no'
WHEN delivery = 1 THEN 'yes'
ELSE 'NULL'
END AS delivery,
CASE WHEN tetanus = 0 THEN 'no'
WHEN tetanus = 1 THEN 'yes'
ELSE 'NULL'
END AS tetanus,
CASE WHEN bcg = 0 THEN 'no'
WHEN bcg = 1 THEN 'yes'
ELSE 'NULL'
END AS bcg,
CASE WHEN measles = 0 THEN 'no'
WHEN measles = 1 THEN 'yes'
ELSE 'NULL'
END AS measles,
CASE WHEN polio = 0 THEN 'no'
WHEN polio = 1 THEN 'yes'
ELSE 'NULL'
END AS polio,
CASE WHEN dpt = 0 THEN 'no'
WHEN dpt = 1 THEN 'yes'
ELSE 'NULL'
END AS dpt,
CASE WHEN chhprob = 0 THEN 'no'
WHEN chhprob = 1 THEN 'yes'
ELSE 'NULL'
END AS chhprob,
CASE WHEN chdisability = 0 THEN 'no'
WHEN chdisability = 1 THEN 'yes'
ELSE 'NULL'
END AS chdisability,

```

```

CASE WHEN chhrel = 1 THEN 'Same'
WHEN chhrel = 2 THEN 'Better'
WHEN chhrel = 3 THEN 'Worse'
ELSE 'NULL'
END AS chhrel,
CASE WHEN chhealth = 1 THEN 'v.poor'
WHEN chhealth = 2 THEN 'poor'
WHEN chhealth = 3 THEN 'average'
WHEN chhealth = 4 THEN 'good'
WHEN chhealth = 5 THEN 'v.good'
ELSE 'NULL'
END AS chhealth,
CASE WHEN preprim = 0 THEN 'no'
WHEN preprim = 1 THEN 'yes'
ELSE 'NULL'
END AS preprim,
    [agegr1],
CASE WHEN enrol = 0 THEN 'no'
WHEN enrol = 1 THEN 'yes'
ELSE 'NULL'
END AS enrol,
CASE WHEN entype = 1 THEN 'private'
WHEN entype = 2 THEN 'public'
WHEN entype = 3 THEN 'community'
WHEN entype = 4 THEN 'gov funded'
ELSE 'NULL'
END AS entype,
CASE WHEN levread = 1 THEN 'cant read'
WHEN levread = 2 THEN 'read letters'
WHEN levread = 3 THEN 'read words'
WHEN levread = 4 THEN 'read sentence'
ELSE 'NULL'
END AS levread,
CASE WHEN levlwrit = 1 THEN 'no'
WHEN levlwrit = 2 THEN 'yes with errors'
WHEN levlwrit = 3 THEN 'yes without errors'
ELSE 'NULL'
END AS levlwrit,
CASE WHEN literate = 0 THEN 'no'

```

```

WHEN literate = 1 THEN 'yes'
ELSE 'NULL'
END AS literate,
    [wi],
    [hq],
CASE WHEN drwaterq = 0 THEN 'no'
WHEN drwaterq = 1 THEN 'yes'
ELSE 'NULL'
END AS drwaterq,
CASE WHEN toiletq = 0 THEN 'no'
WHEN toiletq = 1 THEN 'yes'
ELSE 'NULL'
END AS toiletq,
CASE WHEN elecq = 0 THEN 'no'
WHEN elecq = 1 THEN 'yes'
ELSE 'NULL'
END AS elecq,
CASE WHEN foodsec = 1 THEN 'eat enough of what we want'
WHEN foodsec = 2 THEN 'eat enough but not what we like'
WHEN foodsec = 3 THEN 'sometimes do not eat enough'
WHEN foodsec = 4 THEN 'frequently do not eat enough'
ELSE 'NULL'
END AS foodsec
FROM Ethiopia

```

These views were merged to form one large view with the goal of making queries very simple and short. Script for creating this large view is as follow:

```

---MERGING THE INFORMATION FROM ALL COUNTRIES TO FORM A LARGE VIEW
CREATE VIEW all_data
AS
SELECT * FROM ethiopia_data
UNION ALL
SELECT * FROM vietnam_data
UNION ALL
SELECT * FROM peru_data
UNION ALL
SELECT * FROM india_data

```

## 4.5 REPORTS

To understand child poverty in these four countries, the following reports were created:

- Change in anthropometric characteristics by round
- Child's general health by round
- Literacy by country and round
- Access to basic amenities
- Level of food security

### 4.5.1 CHANGE IN ANTHROPOMETRIC CHARACTERISTICS BY ROUND

This view was created to show the number of children with thinness, stunting and underweight and how this count changes with round across all four countries. The script for creating this view is as follow:

```
--VIEW TO SHOW CHANGES IN ANTHROPOMETRIC CHARACTERISTICS WITH  
ROUND  
CREATE VIEW AnthropometricChanges  
AS  
SELECT a.country, a.round, a.condition, stunting_count,  
thinness_count, underwgt_count  
FROM (  
(select country, round, stunting as condition, count(childid) AS  
stunting_count  
from all_data  
where stunting != 'NULL'  
group by stunting, country, round) AS a  
full outer join  
(select country, round, thinness as condition, count(childid) AS  
thinness_count  
from all_data  
where thinness != 'NULL'  
group by thinness, country, round) AS b  
ON a.condition = b.condition and a.round = b.round and a.country =  
b.country
```

```

full outer join
(select country, round, underweight as condition, count(childid)
AS underwgt_count
from all_data
where underweight != 'NULL'
group by underweight, country, round) AS c
ON c.condition = b.condition and c.round = b.round and c.country =
b.country)

```

The following report can be obtained from this view:

	country	round	condition	stunting_count	thinness_count	underwgt_count
1	ethiopia	1	moderately	603	318	589
2	ethiopia	1	normal	1763	2219	1722
3	ethiopia	1	severely	515	158	457
4	ethiopia	2	moderately	653	366	378
5	ethiopia	2	normal	1991	2371	1454
6	ethiopia	2	severely	238	148	77
7	ethiopia	3	moderately	462	582	514
8	ethiopia	3	normal	2158	2034	1226
9	ethiopia	3	severely	218	216	140
10	ethiopia	4	moderately	468	660	NULL
11	ethiopia	4	normal	1755	1471	NULL
12	ethiopia	4	severely	125	213	NULL
13	ethiopia	5	moderately	324	444	NULL
14	ethiopia	5	normal	1345	1147	NULL
15	ethiopia	5	severely	136	209	NULL
16	india	1	moderately	650	494	780
17	india	1	normal	2039	2354	1868
18	india	1	severely	288	140	353
19	india	2	moderately	822	547	664
20	india	2	normal	1894	2225	1080
21	india	2	severely	198	144	199
22	india	3	moderately	716	603	628
23	india	3	normal	1992	2097	1043
24	india	3	severely	186	186	258
25	india	4	moderately	572	500	NULL

Query executed successfully. | DESKTOP-DLCD2E0\SC

**Figure 3: Change in the number of thin, stunted and underweight children by round**

#### 4.5.2 CHILD'S GENERAL HEALTH BY ROUND

A view was created to show the general health of the children in the four countries by round. Round 1 and 2 showed null all through because there is no record for food security

in round 1 and 2. Various searches can be made on this view depending on a user's preference. The script for creating this view is as follow:

#### --VIEW TO COMPARE CHILDREN GENERAL HEALTH BY COUNTRY

```
CREATE VIEW Childs_GenHealth
AS
SELECT a.country, a.genHealth_condition, round1_count,
round2_count, round3_count, round4_count, round5_count
FROM (
(select country, chhealth as genHealth_condition, count(childid)
AS round3_count
from all_data
where chhealth != 'NULL' and round = 3
group by chhealth, country) AS a
full outer join
(select country, chhealth as genHealth_condition, count(childid)
AS round4_count
from all_data
where chhealth != 'NULL' and round = 4
group by chhealth, country) AS b
ON a.genHealth_condition = b.genHealth_condition and a.country =
b.country
full outer join
(select country, chhealth as genHealth_condition, count(childid)
AS round5_count
from all_data
where chhealth != 'NULL' and round = 5
group by chhealth, country) AS c
ON c.genHealth_condition = b.genHealth_condition and c.country =
b.country
full outer join
(select country, chhealth as genHealth_condition, count(childid)
AS round1_count
from all_data
where chhealth != 'NULL' and round = 1
group by chhealth, country) AS d
ON d.genHealth_condition = c.genHealth_condition and d.country =
c.country
```



```

full outer join
(select country, chhealth as genHealth_condition, count(childid)
AS round2_count
from all_data
where chhealth != 'NULL' and round = 2
group by chhealth, country) AS e
ON e.genHealth_condition = d.genHealth_condition and e.country =
d.country)

```

The report generated from this view is as follow:

	country	genHealth_condition	round1_count	round2_count	round3_count	round4_count	round5_count
1	ethiopia	average	NULL	NULL	503	348	323
2	ethiopia	good	NULL	NULL	1333	992	1112
3	ethiopia	poor	NULL	NULL	112	79	55
4	ethiopia	v.good	NULL	NULL	875	1318	1107
5	ethiopia	v.poor	NULL	NULL	35	42	27
6	india	average	NULL	NULL	386	623	508
7	india	good	NULL	NULL	1993	1786	1849
8	india	poor	NULL	NULL	62	133	95
9	india	v.good	NULL	NULL	443	311	353
10	india	v.poor	NULL	NULL	19	14	11
11	peru	average	NULL	NULL	866	846	705
12	peru	good	NULL	NULL	1449	1488	1487
13	peru	poor	NULL	NULL	39	37	31
14	peru	v.good	NULL	NULL	242	134	213
15	peru	v.poor	NULL	NULL	7	2	5
16	vietnam	average	NULL	NULL	1474	1620	1814
17	vietnam	good	NULL	NULL	1018	830	841
18	vietnam	poor	NULL	NULL	320	274	77
19	vietnam	v.good	NULL	NULL	99	78	110
20	vietnam	v.poor	NULL	NULL	12	11	5

**Figure 4: Count of children based on their general health condition**

### 4.5.3 LITERACY BY COUNTRY AND ROUND

This view was created to show the number of children who are literate compared to those who are illiterate in all four countries. The script for creating this view is as follow:

```
--VIEW SHOWING LITERACY BY COUNTRY AND ROUND
```

```

CREATE VIEW literacy_by_country
AS
SELECT a.country, a.educated, round1_count, round2_count,
round3_count, round4_count, round5_count
FROM (
(select country, literate as educated, count(childid) AS
round3_count
from all_data
where literate != 'NULL' and round = 3
group by literate, country) AS a
LEFT join
(select country, literate as educated, count(childid) AS
round4_count
from all_data
where literate != 'NULL' and round = 4
group by literate, country) AS b
ON a.educated = b.educated and a.country = b.country
LEFT join
(select country, literate as educated, count(childid) AS
round5_count
from all_data
where literate != 'NULL' and round = 5
group by literate, country) AS c
ON c.educated = b.educated and c.country = b.country
LEFT join
(select country, literate as educated, count(childid) AS
round1_count
from all_data
where literate != 'NULL' and round = 1
group by literate, country) AS d
ON d.educated = c.educated and d.country = c.country
LEFT join
(select country, literate as educated, count(childid) AS
round2_count
from all_data
where literate != 'NULL' and round = 2
group by literate, country) AS e
ON e.educated = d.educated and e.country = d.country)

```

The following report was obtained from this view:

	country	educated	round1_count	round2_count	round3_count	round4_count	round5_count
1	ethiopia	no	2862	2561	2712	2999	2999
2	ethiopia	yes	NULL	NULL	287	NULL	NULL
3	india	no	2649	2379	2317	3019	3019
4	india	yes	NULL	NULL	702	NULL	NULL
5	peru	no	2399	2190	1600	2766	2766
6	peru	yes	NULL	NULL	1166	NULL	NULL
7	vietnam	yes	NULL	NULL	1624	NULL	NULL
8	vietnam	no	2271	2092	1376	3000	3000

**Figure 5: Literacy level by round**

#### 4.5.4 ACCESS TO BASIC AMENITIES

This view was created to show the number of children who has access to basic amenities such as (portable drinking water, electricity and toilet) compared to those who does not and how this count changes with round across all four countries. The script for creating this view is as follow:

```
/* CREATE A VIEW SEE THE NUMBER OF CHILDREN WHO HAVE ACCESS TO
DRINKING WATER,
ELECTRICITY AND TOILET*/
```

```
CREATE VIEW AccessToAmenities
AS
SELECT a.country, a.round, a.access_to_amenity, Toilet_count,
DrinkingWater_count, Electricity_count
FROM (
(select country, round, toiletq as access_to_amenity,
count(childid) AS Toilet_count
from all_data
where toiletq != 'NULL'
group by toiletq, country, round) AS a
full outer join
(select country, round, drwaterq as access_to_amenity,
count(childid) AS DrinkingWater_count
from all_data
where drwaterq != 'NULL'
```

```

group by drwaterq, country, round) AS b
ON a.access_to_amenity = b.access_to_amenity and a.round = b.round
and a.country = b.country
full outer join
(select country, round, elecq as access_to_amenity, count(childid)
AS Electricity_count
from all_data
where elecq != 'NULL'
group by elecq, country, round) AS c
ON c.access_to_amenity = b.access_to_amenity and c.round = b.round
and c.country = b.country)

```

The following report is obtained from this view:

	country	round	access_to_amenity	Toilet_count	DrinkingWater_count	Electricity_count
1	ethiopia	1	no	1872	1371	1935
2	ethiopia	1	yes	1126	1623	1062
3	ethiopia	2	yes	1587	1722	1264
4	ethiopia	2	no	1303	1170	1628
5	ethiopia	3	yes	2104	1746	1422
6	ethiopia	3	no	755	1112	1436
7	ethiopia	4	yes	2143	1695	1752
8	ethiopia	4	no	638	1087	1030
9	ethiopia	5	no	622	760	888
10	ethiopia	5	yes	2039	1901	1773
11	india	1	no	2099	497	540
12	india	1	yes	917	2522	2479
13	india	2	yes	981	2798	2638
14	india	2	no	1963	146	306
15	india	3	no	1891	91	95
16	india	3	yes	1012	2815	2811
17	india	4	yes	1215	2829	2792
18	india	4	no	1652	38	75
19	india	5	yes	1486	2809	2782
20	india	5	no	1351	28	55
21	peru	1	yes	2166	1424	1863
22	peru	1	no	600	1342	903
23	peru	2	yes	2263	1653	2039
24	peru	2	no	385	995	609
25	peru	3	yes	2386	2123	2275

**Figure 6: Access to Basic Amenities**

#### 4.5.5 FOOD SECURITY

This view was created to show the count of children based on their level of food security in round 3 to 4 in the four countries. Round 1 and 2 was not included in this view as there is no record for food security in round 1 and 2. The script for creating this view is as follow:

```
--CREATE A VIEW TO SHOW CHILDREN LEVEL OF FOOD SECURITY BY ROUND
```

```
CREATE VIEW foodsecurity
AS
SELECT a.country, a.foodsecurity as foodsecurity, round3_count,
round4_count, round5_count
FROM (
(select country, foodsec as foodsecurity, count(childid) AS
round3_count
from all_data
where foodsec != 'NULL' and round = 3
group by foodsec, country
HAVING foodsec != 'NULL') AS a
full outer join
(select country, foodsec as foodsecurity, count(childid) AS
round4_count
from all_data
where foodsec != 'NULL' and round = 4
group by foodsec, country
HAVING foodsec != 'NULL') AS b
ON a.foodsecurity = b.foodsecurity and a.country = b.country
full outer join
(select country, foodsec as foodsecurity, count(childid) AS
round5_count
from all_data
where foodsec != 'NULL' and round = 5
group by foodsec, country
HAVING foodsec != 'NULL') AS c
ON c.foodsecurity = b.foodsecurity and c.country = b.country)
```

	country	foodsecurity	round3_count	round4_count	round5_count
1	ethiopia	eat enough but not what we like	1545	1079	1128
2	ethiopia	eat enough of what we want	264	185	176
3	ethiopia	frequently do not eat enough	168	61	55
4	ethiopia	sometimes do not eat enough	882	549	460
5	india	eat enough but not what we like	1805	1166	1118
6	india	eat enough of what we want	825	667	702
7	india	frequently do not eat enough	37	5	15
8	india	sometimes do not eat enough	237	75	74
9	peru	eat enough but not what we like	1582	940	933
10	peru	eat enough of what we want	761	752	721
11	peru	frequently do not eat enough	42	23	21
12	peru	sometimes do not eat enough	220	170	162
13	vietnam	eat enough but not what we like	1493	683	797
14	vietnam	eat enough of what we want	546	962	907
15	vietnam	frequently do not eat enough	157	74	49
16	vietnam	sometimes do not eat enough	730	210	186

**Figure 7: Food security**

## 4.6 STORED PROCEDURES

The following stored procedures were written to help users search for specific information they are interested in from the database:

### Change in anthropometric characteristics by round:

A parameterized stored procedure was created to help users filter the result of the view created to show change in anthropometric characteristics to suit their preference. Script for creating this stored procedure is as follow:

```
--GETTING THE ANTHROPOMETRIC REPORT FOR A PARTICULAR COUNTRY AND  
CONDITION
```

```
CREATE PROCEDURE uspAnthropometricChar  
(  
    @country AS NVARCHAR(100)  
    ,@condition AS NVARCHAR(100)  
)  
AS  
BEGIN  
    SELECT *  
    FROM  
        AnthropometricChanges
```

```

WHERE
    country = @country AND
    condition LIKE @condition + '%'
ORDER BY round;
END;

```

Suppose a user is interested in know the number of children in a particular country having a specific anthropometric characteristics remark the above stored procedure can be executed as follow:

```

--EXECUTING THE STORED PROCEDURE
EXEC uspAnthropometricChar peru, severe;

```

	country	round	condition	stunting_count	thinness_count	underwgt_count
1	peru	1	severely	222	19	33
2	peru	2	severely	212	2	13
3	peru	3	severely	112	10	16
4	peru	4	severely	82	3	NULL
5	peru	5	severely	37	4	NULL

**Figure 8: Showing the count of children who has severe stunting, thinness and underweight condition in Peru**

### Child's general health by round

Suppose a user is interested in knowing the general health of children in a particular country, the following stored procedure can be used:

```

/*GETTING THE GENERAL HEALTH CONDITION REPORT FOR CHILDREN
IN A PARTICULAR COUNTRY AND CONDITION*/

```

```

CREATE PROCEDURE uspGenHealthByCountry
(
    @country AS NVARCHAR(100)
    ,@condition AS NVARCHAR(100)
)
AS

```



```

BEGIN
    SELECT *
    FROM
        Childs_GenHealth
    WHERE
        country = @country AND
        genHealth_condition = @condition
    ORDER BY country, genHealth_condition;
END;

```

--EXECUTING THE STORED PROCEDURE

```
EXEC uspGenHealthByCountry ethiopia, good;
```

Results		Messages					
	country	genHealth_condition	round1_count	round2_count	round3_count	round4_count	round5_count
1	ethiopia	good	NULL	NULL	1333	992	1112

**Figure 9: Show the count of Children with good general health condition in Ethiopia by round**

### Literacy by country and round

Suppose a user is rather interested in knowing the literacy rate for just one country by gender, the following stored procedure can be executed:

```

/*GETTING THE LITERACY RATE BY COUNTRY AND GENDER*/
CREATE PROCEDURE uspLiteracyRate
(
    @country AS NVARCHAR(100)
)
AS
BEGIN
    SELECT a.country, a.chsex, a.educated, round1_count,
    round2_count, round3_count, round4_count, round5_count
    FROM (
        (select country, chsex, literate as educated,
        count(childid) AS round3_count
        from all_data

```



```

where literate != 'NULL' and round = 3
group by literate, country, chsex) AS a
LEFT join
(select country, chsex, literate as educated, count(childid)
AS round4_count
from all_data
where literate != 'NULL' and round = 4
group by literate, country, chsex) AS b
ON a.educated = b.educated and a.country = b.country and
a.chsex = b.chsex
LEFT join
(select country, chsex, literate as educated, count(childid)
AS round5_count
from all_data
where literate != 'NULL' and round = 5
group by literate, country, chsex) AS c
ON c.educated = b.educated and c.country = b.country and
c.chsex = b.chsex
LEFT join
(select country, chsex, literate as educated, count(childid)
AS round1_count
from all_data
where literate != 'NULL' and round = 1
group by literate, country, chsex) AS d
ON d.educated = c.educated and d.country = c.country and
d.chsex = b.chsex
LEFT join
(select country, chsex, literate as educated, count(childid)
AS round2_count
from all_data
where literate != 'NULL' and round = 2
group by literate, country, chsex) AS e
ON e.educated = d.educated and e.country = d.country and
e.chsex = b.chsex)
WHERE
a.country = @country
ORDER BY country;
END;

```

--EXECUTING THE STORED PROCEDURE

```
EXEC uspLiteracyRate peru;
```

	country	chsex	educated	round1_count	round2_count	round3_count	round4_count	round5_count
1	peru	male	no	1224	1110	843	1413	1413
2	peru	female	no	1175	1080	757	1353	1353
3	peru	female	yes	NULL	NULL	596	NULL	NULL
4	peru	male	yes	NULL	NULL	570	NULL	NULL

**Figure 10: Showing Literacy Rate in Peru by Gender**

### Access to basic amenities

Suppose a user is only interested in comparing the count of children who has access to basic amenities across the four countries the following stored procedure can be used:

```
CREATE PROCEDURE uspAccessToAmenity
(
    @accessToAmenity AS NVARCHAR(100)
)
AS
BEGIN
    SELECT *
    FROM
        AccessToAmenities
    WHERE
        access_to_amenity = @accessToAmenity
    ORDER BY country;
END;
```

--EXECUTING THE STORED PROCEDURE

```
EXEC uspAccessToAmenity yes;
```

```
EXEC uspAccessToAmenity no;
```

	country	round	access_to_amenity	Toilet_count	DrinkingWater_count	Electricity_count
1	ethiopia	3	yes	2104	1746	1422
2	ethiopia	4	yes	2143	1695	1752
3	ethiopia	1	yes	1126	1623	1062
4	ethiopia	2	yes	1587	1722	1264
5	ethiopia	5	yes	2039	1901	1773
6	india	5	yes	1486	2809	2782
7	india	2	yes	981	2798	2638
8	india	1	yes	917	2522	2479
9	india	4	yes	1215	2829	2792
10	india	3	yes	1012	2815	2811
11	peru	1	yes	2166	1424	1863
12	peru	4	yes	2384	2048	2392
13	peru	2	yes	2263	1653	2039
14	peru	5	yes	2356	2024	2362
15	peru	3	yes	2386	2123	2275
16	vietnam	4	yes	2158	740	2797

**Figure 11: Showing the count Children who has access to basic amenities**

### Level of food security

Suppose a user is only interesting in comparing record of children who has a specific level of food security, the following stored procedure can be executed

```
--CREATE STORE PROCEDURE TO SHOW SPECIFIC SEARCH CONDITION
CREATE PROCEDURE uspfoodsecurity
(
    @foodsec AS NVARCHAR(100)
)
AS
BEGIN
    SELECT *
    FROM
        foodsecurity
    WHERE
        foodsecurity = @foodsec
    ORDER BY country;
END;
```

```
--EXECUTE STORED PROCEDURE
```

```
EXEC uspfoodsecurity [sometimes do not eat enough];
```

Results		Messages			
	country	foodsecurity	round3_count	round4_count	round5_count
1	ethiopia	sometimes do not eat enough	882	549	460
2	india	sometimes do not eat enough	237	75	74
3	peru	sometimes do not eat enough	220	170	162
4	vietnam	sometimes do not eat enough	730	210	186

**Figure 12: Showing the count of children who lack food to eat**

## 5.0 DATABASE SECURITY AND BACKUP

This database is susceptible to the following security threats:

- Theft and fraud
- Loss of privacy
- Loss of integrity
- Loss of confidentiality
- Loss of availability

To mitigate these threats the following should be implemented:

- Authorization controls should be implemented to control users' access to database objects. This can be done by authenticating users requesting to access the database objects such as tables, views etc. So users of the database should have login details peculiar to only them and must go through the authentication process before they have access to database objects.
- Permission to write, read, update database objects should be granted to users according to their roles. The need to update this database only arises after a round so the database is not to be updated often and this can be single handedly handled by the database administrator. So for this database, the outright privilege to create,

read, write and modify database objects should only be granted to the database administrator. All other users should be granted just the privilege to read some or all database base objects depending on their tasks or job requirements.

- View should be used to limit the information certain users have access to read in order to hide part of the database from some users.
- To prevent loss of data backup should be taken periodically. Since the database is not dynamic, full backup should be taken every six months and after every update is made on the database. Differential backup should be taken every week and log backup should be taken every day to keep track of transaction logs occurring in the database.

## **5.1 DATA PRIVACY, ETHICAL AND LEGAL ISSUES**

Information stored on this database is personal information and as such it must comply with the eight principles that govern personal data handling contained in UK Data Protection Act 1998 and The Freedom of Information Act 2000. These principles are:

1. Fairly and lawfully processed
2. Processed for limited purposes
3. Adequate, relevant and not excessive
4. Accurate and up to date
5. Not kept for longer than is necessary
6. Processed in line with your rights
7. Secure
8. Not transferred to other countries without adequate protection

To ensure compliance with to the Data Protection Act, the following must be adhered to strictly:

- Children, parents, caregivers and every individual directly involved in a child's life must consent to the collection and storage of this vital information about the child.
- Information collected should not be used only for a specific legal purpose
- Only information that are imperative to the organization should be collected about the child.
- The organization should always ensure that the information collected is correct and up-to-date.
- The data should be kept securely and the organization should implement access control and authentication to control access to the data.
- In case there is any breach or unauthorized access on the database, it should be reported to the relevant authorities timely.

## **6.0 CONCLUSION**

Child well being monitor is a simple reporting system that was designed using data gotten from the young lives survey which was aimed at understanding child poverty in lower income Countries. It can be used to produce various summarized reports to understand the effects of child poverty on the education, health and general living condition of children living below the minimum acceptable standard of living.

This proposal documents the procedures and T-SQL statements used in designing this reporting tool and various parameterized stored procedures that can be used to filter the reports to suit the users search requirements.