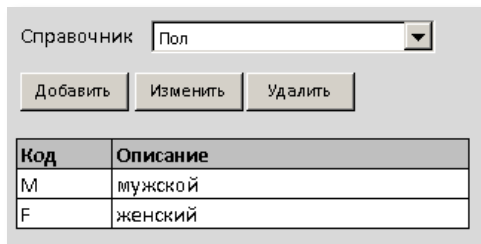


Вариант хранения множества простых справочников в 2-х таблицах

Данный подход был использован мною в одном проекте для хранения множества простых справочников содержащих небольшое количество записей вида «код, значение».

Думаю, данная статья в основном будет интересна начинающим разработчикам.

Преимущество данного метода в том, что для ввода всех справочников достаточно создать одну форму с выпадающим списком, который служит для переключения между справочниками и таблицей для ввода/редактирования значений выбранного справочника:



Код	Описание
М	мужской
F	женский

Реализация на примере БД Oracle

Было создано 2 таблицы **SIMPLE_REF** (описание справочников) и **SIMPLE_REF_VAL** (значения справочников):

-- таблица для описание справочников

```
CREATE TABLE SIMPLE_REF (  
  ID      VARCHAR2(3 CHAR) NOT NULL PRIMARY KEY,  
  DESCR  VARCHAR2(500 CHAR)  
);  
  
COMMENT ON TABLE SIMPLE_REF IS 'Простые справочники';  
COMMENT ON COLUMN SIMPLE_REF.ID IS 'ID справочника';  
COMMENT ON COLUMN SIMPLE_REF.DESCR IS 'Описание';
```

-- примеры значений

```
INSERT INTO SIMPLE_REF(ID,DESCR) VALUES ('SEX', 'Пол');  
INSERT INTO SIMPLE_REF(ID,DESCR) VALUES ('CNT', 'Страна');
```

-- таблица для хранения значений справочников

```
CREATE TABLE SIMPLE_REF_VAL (  
  ID      VARCHAR2(10 CHAR) NOT NULL PRIMARY KEY,  
  REF_ID  VARCHAR2(3 CHAR) NOT NULL,  
  CODE   VARCHAR2(10 CHAR) NOT NULL,  
  DESCR  VARCHAR2(500 CHAR),  
  -- данное ограничение будет гарантировать, что ID будут с учетом префикса  
  CONSTRAINT SIMPLE_REF_VAL_ID_CH CHECK(ID LIKE REF_ID||'%'),  
  CONSTRAINT SIMPLE_REF_VAL_REF_ID_FK FOREIGN KEY(REF_ID) REFERENCES SIMPLE_REF(ID),  
  CONSTRAINT SIMPLE_REF_VAL_REF_ID_CODE_UN UNIQUE(REF_ID, CODE)  
);
```

-- в случае отсутствия поля CODE и SIMPLE_REF_VAL_REF_ID_CODE_UN создаем индекс по REF_ID

```
--CREATE INDEX SIMPLE_REF_VAL_REF_ID_IDX ON SIMPLE_REF_VAL(REF_ID);
```

```
COMMENT ON TABLE SIMPLE_REF_VAL IS 'Значения простых справочников';  
COMMENT ON COLUMN SIMPLE_REF_VAL.ID IS 'ID значения';  
COMMENT ON COLUMN SIMPLE_REF_VAL.REF_ID IS 'ID справочника';  
COMMENT ON COLUMN SIMPLE_REF_VAL.DESCR IS 'Код для вывода';  
COMMENT ON COLUMN SIMPLE_REF_VAL.DESCR IS 'Описание значения';
```

-- примеры значений для справочника SEX

```
INSERT INTO SIMPLE_REF_VAL(ID,REF_ID,CODE,DESCR) VALUES ('SEX0000001', 'SEX', 'M', 'мужской');  
INSERT INTO SIMPLE_REF_VAL(ID,REF_ID,CODE,DESCR) VALUES ('SEX0000002', 'SEX', 'F', 'женский');
```

-- примеры значений для справочника CNT

```
INSERT INTO SIMPLE_REF_VAL(ID,REF_ID,CODE,DESCR) VALUES ('CNTABH', 'CNT', 'ABH', 'Абхазия');  
INSERT INTO SIMPLE_REF_VAL(ID,REF_ID,CODE,DESCR) VALUES ('CNTAUS', 'CNT', 'AUS', 'Австралия');  
INSERT INTO SIMPLE_REF_VAL(ID,REF_ID,CODE,DESCR) VALUES ('CNTAUT', 'CNT', 'AUT', 'Австрия');  
INSERT INTO SIMPLE_REF_VAL(ID,REF_ID,CODE,DESCR) VALUES ('CNTAZE', 'CNT', 'AZE', 'Азербайджан');  
INSERT INTO SIMPLE_REF_VAL(ID,REF_ID,CODE,DESCR) VALUES ('CNTALB', 'CNT', 'ALB', 'Албания');
```

Для генерации новых значений используется сиквенс:

```
CREATE SEQUENCE SIMPLE_REF_VAL_SEQ START WITH 100;
```

Для генерации нового значения ID используется запрос, в котором при помощи параметра **pRefID** передается ID справочника (SEX, CNT):

```
SELECT RPAD(:pRefID,3,' ')||TRIM(TO_CHAR(SIMPLE_REF_VAL_SEQ.NEXTVAL,'000000')) GEN_ID
FROM DUAL
```

В общем виде добавление выглядит следующим образом:

Справочник: Пол

Добавить | Изменить | Удалить

Код | Описание

M	мужской
F	женский

```
:pRefID = 'SEX'
SELECT RPAD(:pRefID,3,' ')||TRIM(TO_CHAR(SIMPLE_REF_VAL_SEQ.NEXTVAL,'000000')) GEN_ID
FROM DUAL
ID = 'SEX0000100'
```

Пример таблицы использующей значения справочников:

```
CREATE TABLE PERSON(
  ID      NUMBER(*, 0) NOT NULL PRIMARY KEY,
  FNAME  VARCHAR2(50 CHAR) NOT NULL,
  LNAME  VARCHAR2(50 CHAR) NOT NULL,
  -- поле с проверкой, что ID начинается с SEX
  SEX_ID VARCHAR2(10 CHAR) NOT NULL CHECK(SEX_ID LIKE 'SEX%'),
  -- поле с проверкой, что ID начинается с CNT
  CNT_ID VARCHAR2(10 CHAR) NOT NULL CHECK(CNT_ID LIKE 'CNT%'),
  -- ссылки на таблицу SIMPLE_REF_VAL
  CONSTRAINT PERSON_SEX_ID_FK FOREIGN KEY(SEX_ID) REFERENCES SIMPLE_REF_VAL(ID),
  CONSTRAINT PERSON_CNT_ID_FK FOREIGN KEY(CNT_ID) REFERENCES SIMPLE_REF_VAL(ID)
);
```

```
COMMENT ON TABLE PERSON IS 'Карточки лиц';
COMMENT ON COLUMN PERSON.ID IS 'ID';
COMMENT ON COLUMN PERSON.FNAME IS 'Имя';
COMMENT ON COLUMN PERSON.LNAME IS 'Фамилия';
COMMENT ON COLUMN PERSON.SEX_ID IS 'Пол';
COMMENT ON COLUMN PERSON.CNT_ID IS 'Страна';
```

-- пример значения

```
INSERT INTO PERSON(ID, FNAME, LNAME, SEX_ID, CNT_ID) VALUES (1, 'Иван', 'Иванов', 'SEX0000001', 'CNTABH');
```

Использование ограничений вида «**CHECK(XXX_ID LIKE 'XXX%')**» будет гарантировать, что в поле могут быть записаны только значения из соответствующего справочника.

Выборка необходимых значений из справочников, например, для формирования выпадающих списков:

ID: 1

Фамилия: Иванов

Имя: Иван

Пол: мужской

Страна: Абхазия

```
SELECT * FROM SIMPLE_REF_VAL WHERE REF_ID='SEX'
SELECT * FROM SIMPLE_REF_VAL WHERE REF_ID='CNT'
```

Заключение

Данный подход имеет как свои преимущества, так и недостатки, один из недостатков – снижение скорости при получении описаний из справочников, например:

```
SELECT P.*, SEX.DESCR SEX_DESCR, CNT.DESCR CNT_DESCR
FROM PERSON P
JOIN SIMPLE_REF_VAL SEX ON P.SEX_ID=SEX.ID
JOIN SIMPLE_REF_VAL CNT ON P.CNT_ID=CNT.ID
```

Если значений в каких-то справочниках очень много, то возможно лучше выделить их в отдельные таблицы. Удачных проектов!