Old online shopping logical model: Customers, Orders, Suppliers
New book managing logical model: Student, Book, Borrow Date

<table>
<thead>
<tr>
<th>Table</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book</td>
<td>P: Book_ID, Title, EID</td>
</tr>
<tr>
<td></td>
<td>F: Book_ID, Author_ID,</td>
</tr>
<tr>
<td></td>
<td>Date_ID, Student_ID</td>
</tr>
<tr>
<td></td>
<td>BookAuthor PK (BA_ID)</td>
</tr>
<tr>
<td>BookAuthor</td>
<td>P: Author_ID, Name</td>
</tr>
<tr>
<td></td>
<td>F: Book_ID, Author_ID</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Author</td>
<td>P: Author_ID, Name</td>
</tr>
<tr>
<td></td>
<td>F: Author_ID</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Student</td>
<td>P: Student_ID, EID</td>
</tr>
<tr>
<td></td>
<td>F: Classification_ID,</td>
</tr>
<tr>
<td></td>
<td>Classification_ID,</td>
</tr>
<tr>
<td></td>
<td>Date_ID, Student_ID</td>
</tr>
<tr>
<td></td>
<td>Student PK (Student_ID)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>P: Date_ID, Date</td>
</tr>
<tr>
<td></td>
<td>F: Date_ID</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>SBD</td>
<td>P: SBD_ID, Integer</td>
</tr>
<tr>
<td></td>
<td>F: SBD_ID, Integer</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>International Citizen</td>
<td>Country, VARCH (256)</td>
</tr>
<tr>
<td>U.S. Citizen</td>
<td>State, VARCH (256)</td>
</tr>
</tbody>
</table>
CREATE TABLE SBD ( SBD_ID INTEGER NOT NULL, Book_ID INTEGER NOT NULL, Date_ID INTEGER NOT NULL, Student_ID INTEGER NOT NULL )
ALTER TABLE SBD ADD CONSTRAINT SBD_PK PRIMARY KEY ( SBD_ID );

CREATE SEQUENCE Book_Book_ID_SEQ START WITH 1 NOCACHE ORDER; CREATE OR REPLACE TRIGGER Book_Book_ID_TRG BEFORE INSERT ON Book FOR EACH ROW WHEN (NEW.Book_ID IS NULL) BEGIN :NEW.Book_ID := Book_Book_ID_SEQ.NEXTVAL; END; /

CREATE SEQUENCE BookAuthor_BA_ID_SEQ START WITH 1 NOCACHE ORDER; CREATE OR REPLACE TRIGGER BookAuthor_BA_ID_TRG BEFORE INSERT ON BookAuthor FOR EACH ROW WHEN (NEW.BA_ID IS NULL) BEGIN :NEW.BA_ID := BookAuthor_BA_ID_SEQ.NEXTVAL; END; /

CREATE SEQUENCE Classification_Classification_ID_SEQ START WITH 1 NOCACHE ORDER; CREATE OR REPLACE TRIGGER Classification_Classification_ID_TRG BEFORE INSERT ON Classification FOR EACH ROW WHEN (NEW.Classification_ID IS NULL) BEGIN :NEW.Classification_ID := Classification_Classification_ID_SEQ.NEXTVAL; END; /

CREATE SEQUENCE Date_Date_ID_SEQ START WITH 1 NOCACHE ORDER; CREATE OR REPLACE TRIGGER Date_Date_ID_TRG BEFORE INSERT ON "Date" FOR EACH ROW WHEN (NEW.Date_ID IS NULL) BEGIN :NEW.Date_ID := Date_Date_ID_SEQ.NEXTVAL; END; /

CREATE SEQUENCE SBD_SBD_ID_SEQ START WITH 1 NOCACHE ORDER; CREATE OR REPLACE TRIGGER SBD_SBD_ID_TRG BEFORE INSERT ON SBD FOR EACH ROW WHEN (NEW.SBD_ID IS NULL) BEGIN :NEW.SBD_ID := SBD_SBD_ID_SEQ.NEXTVAL; END; /

CREATE SEQUENCE Student_Student_ID_SEQ START WITH 1 NOCACHE ORDER; CREATE OR REPLACE TRIGGER Student_Student_ID_TRG BEFORE INSERT ON Student FOR EACH ROW WHEN (NEW.Student_ID IS NULL) BEGIN :NEW.Student_ID := Student_Student_ID_SEQ.NEXTVAL; END; /
-- disable FK constraints
alter table Student disable constraint "R1";
alter table SBD disable constraint "R2";
alter table SBD disable constraint "R3";
alter table SBD disable constraint "R4";
alter table BookAuthor disable constraint "R5";
alter table BookAuthor disable constraint "R6";

-- Truncate tables
truncate table Classification;
truncate table Student;
truncate table SBD;
truncate table BookAuthor;
truncate table Book;
truncate table Date;
truncate table Author;

-- insert Classification
insert into Classification(Classification_id, year) values(1, 1);
insert into Classification(Classification_id, year) values(2, 2);
insert into Classification(Classification_id, year) values(3, 3);

-- insert Student
insert into Student(Student_ID, Classification_id, EID, type, Country, State) values(1, 1, 'aa233', 'U.S. Citizen', null, 'Texas')
insert into Student(Student_ID, Classification_id, EID, type, Country, State) values(2, 1, 'ii233', 'International Citizen', 'Antarctica', null)
insert into Student(Student_ID, Classification_id, EID, type, Country, State) values(3, 1, 'op233', 'Student', null, null)

-- insert Book
insert into Book(Book_ID, Title) values(1, 'Cosmic Prophet');
insert into Book(Book_ID, Title) values(2, 'Hydro');
insert into Book(Book_ID, Title) values(3, 'Teos');

-- insert Date
insert into Date(Date_ID, Date) values(1, '09-30-2016');
insert into Date(Date_ID, Date) values(2, '09-23-2016');
insert into Date(Date_ID, Date) values(3, '09-24-2016');

-- insert SBD
insert into SBD(SBD_ID, Book_ID, Date_ID, Student_ID) values(1, 1, 1, 1);
insert into SBD(SBD_ID, Book_ID, Date_ID, Student_ID) values(2, 2, 2, 2);
insert into SBD(SBD_ID, Book_ID, Date_ID, Student_ID) values(3, 3, 3, 3);

-- insert author
insert into author(author_id, name) values(1, 'Laren');
insert into author(author_id, name) values(2, 'Feeny');
insert into author(author_id, name) values(3, 'Joe');

-- insert BookAuthor
insert into BookAuthor(BA_ID, Book_ID, author_id) values(1, 1, 1);
insert into BookAuthor(BA_ID, Book_ID, author_id) values(2, 2, 2);
insert into BookAuthor(BA_ID, Book_ID, author_id) values(3, 3, 3);

-- Enable FK constraints
alter table Student enable constraint "R1";
alter table SBD enable constraint "R2";
alter table SBD enable constraint "R3";
alter table SBD enable constraint "R4";
alter table BookAuthor enable constraint "R5";
alter table BookAuthor enable constraint "R6";
APEX Application

Classification Select List
select year as d,
       classification_id as r
from classification
order by 1

Student Interactive Report SQL
select "STUDENT_ID",
    "CLASSIFICATION_ID",
    "EID",
    "TYPE",
    "COUNTRY",
    "STATE"
from "#OWNER#"."STUDENT"
Ternary Relationship Select Lists

**Book_ID**

```sql
select title as d, book_id as r
from book
where book_id not in (select book_id from SBD where student_id = :P14_student_id)
order by 1
```

**Date**

```sql
select "Date" as d, DATE_ID as r
from "Date"
order by 1
```

**Student**

```sql
select eid as d, student_id as r
from student
order by 1
```
Foreign Key Eliminations

Book Author Page

select "BA_ID",
"TITLE",
"NAME"
from "#OWNER#"."BOOKAUTHOR" ba
join "#OWNER#"."BOOK" bk on
ba.book_id = bk.book_id
join "#OWNER#"."AUTHOR" au on
ba.author_id = au.author_id

SBD Page

select "SBD_ID",
"TITLE",
"Date",
"EID"
from "#OWNER#"."SBD" sbd
join "#OWNER#"."BOOK" bk on sbd.book_id =
bk.book_id
join "#OWNER#"."Date" dt on sbd.date_id =
dt.date_id
join "#OWNER#"."STUDENT" st on
sbd.student_id = st.student_id
Chart Drill-Down

**Master**

```sql
select null as URL, classification_id, count (*) as StuCount
from Student
group by Classification_id
order by Classification_id
```

**Detail**

```sql
select Classification_ID, eid
from Student
where classification_id = :P19_CLASSIFICATION_ID
```
Master-Detail Report

**Master**

```sql
select title, count(s.student_id) as StuCount, bk.book_id
from sbd s right join book bk
on s.book_id = bk.book_id
on s.book_id = bk.book_id
group by bk.title, bk.book_id
order by bk.title
```

**Detail**

```sql
select eid, classification_id
from student st join sbd
on st.student_id = sbd.student_id
where book_id = :P17_BOOK_ID
```