The Advantages of Using CHECK Constraints in The Academic Database Tables

Ema Utami
Magister of Informatics Engineering of The College of Information Systems and Computer Science AMIKOM, Ring Road Utara ST, Condong Catur, Depok Sleman Yogyakarta Indonesia
Email: ema.u@amikom.ac.id

Abstract— **Database plays important role on both web-based or desktop-based information system. Nowadays web-based information systems dominate information system application. Relational database systems such as Oracle, MySQL, MS SQL Server or PostgreSQL are familiarly used as data base management system. In Indonesia almost all academic information systems at higher education institutions are web-based application with RDBMS as database software. Some researches show the lack of data input validation that makes problem on data security and data quality. Input sanity is a problem that needs a serious attention. Data validation can be done by making a constraint in the database or application level. The fact is that validation in database level is less done by programmers. This research will perform the analysis of advantages or disadvantages from CHECK constraint on database table of academic information system.**

Index Terms— **CHECK constraint, input validation, input benchmark, data quality, academic database**

I. Introduction

DATABASE is the heart of information system, information system without a database will suffer from many problems. A well-designed and well-managed database is an extremely valuable tool in supporting decision making [1]. Nowadays most information systems use relational database management system (RDBMS) as a data storage software. Oracle, MS SQL Server, PostgreSQL or MySQL are commonly used RDBMS software. Almost all academic information system (AIS) in Indonesian higher education institutions (HEI) are web based application with RDBMS as database software.

II. Related Work

AIS database may contain hundreds of tables having complex relationships and have problems in designing that guarantees integrity and consistency among subsystems [2]. Many studies on database in Indonesia discuss the development of informatics system by relational database where the database program is still limited in using PRIMARY KEY, FOREIGN KEY, NOT NULL and data type as constraints database [3]. Using of only those constraints lead database to face data integrity problems. Before implementing the database, people spent tremendous amounts of time designing the correct database storage and then just left the data unprotected or less treated like buckets that would accept everything [4]. New software-engineering methods allow engineers to create a high number of different programs tailored to the customer needs from common code base but usually do not take the database schema into account [5].

Web-based applications as widely used in academic information systems are being targeted for attackers that exploit XSS and SQL injection vulnerabilities. One of the promising approaches to prevent the exploitation of these vulnerabilities is robust, automated sanitization of untrusted input [6]. One of the methods to prevent data input mistakes is input validation at the database level by using CHECK constraint, but it is less familiar. Most programmers prefer doing input validation at the application level using programming languages such as PHP, Delphi, Visual Basic or other programming languages. Does the use of CHECK constraint have disadvantages to validate data? This research will analyze advantages and disadvantages of CHECK constraint in order to validate the data input to a database table.

III. Methodology and Experiments

To able to answer what the advantages and disadvantages of CHECK constraint, it is necessary to do experimental on a database system having software and hardware specification :

1) Processor AMD Athlon (tm) II X3 455
2) RAM 4 GB
3) Hard disk 250 GB
4) Operating System Linux Debian 6.0
5) PostgreSQL 9.1
6) PHP 5.3.3
7) Apache 2.2.16

This research uses data source from the AIS database of STMIK AMIKOM Yogyakarta, one of HEIs in Indonesia. The research is divided into 2 experiments by using single database called *researchdb*. In the first experiment, the research will test on how long to perform the INSERT and COPY statement on table with and without CHECK constraint. The second experiment, the research will test on how to find CHECK constraint advantages and disadvantages using application program. In the next sections, the research explains the applied process to collect information on advantages and disadvantages of CHECK constraint.

The first experiment uses a single table named *student* created on *researchdb* database and it has single

© 2014 ACADEMY PUBLISHER
doi:10.4304/jsw.9.2.382-388

constraint, PRIMARY KEY. The input data process will perform on that table and record how long the INSERT and COPY statements perform. After the first test has finished the table will be dropped and replaced by a new table having 4 constraints, a PRIMARY KEY and 3 additional CHECK constraints. The same process will be done on this table. The experiment flow diagram is figured on Figure 1.

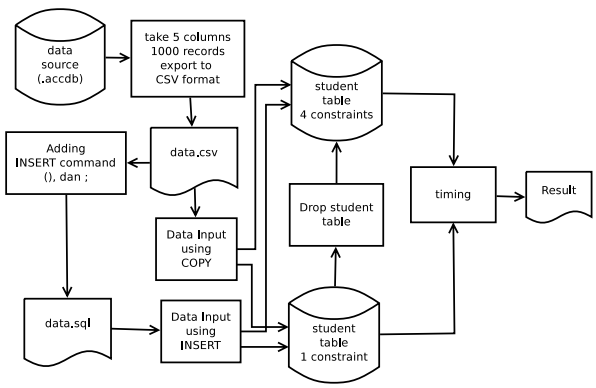

Figure 1. Experiment Flow Diagram

The student table has 5 columns: *name, place of birth (pob), date of birth (dob), sex* and *student id (sid)*. The student table descriptions are shown on Table I.

TABLE I.
STUDENT TABLE

student	
◦name	VARCHAR(40)
◦pob	VARCHAR(25)
◦dob	DATE
◦sex	CHAR
♦sid	CHAR(10)

The table I has single constraint, a PRIMARY KEY on *sid* column and has CHARACTER data type on *name, pob, sex,* and *sid* columns and DATE data type on *dob* column. The SQL command to create a table I for the first testing is shown on Figure 2.

```
CREATE TABLE student (
name VARCHAR(40),
pob VARCHAR(25),
dob DATE,
sex CHAR,
sid CHAR(10) PRIMARY KEY);
```

Figure 2. CREATE statement on a single constraint table

The second testing is conducted by using the same table description but CHECK constraints are added into *dob, sex* and *sid* columns. The CHECK constraint is one of the constraints defined by ANSI SQL used to validate when data are inputted into a table [7]. A CHECK constraint is the most generic constraint type. It allows to specify that the value in a certain column must satisfy a Boolean (truth-value) expression [8]. CHECK constraints are fast, row-level integrity checks offered by most database servers as a standard, built-in RDBMS tool [9]. CHECK constraints are useful for ensuring the enforcement of general data-validation rules or simple business rules [10]. Description of student table for this testing is shown in Table II.

TABLE II.
STUDENT TABLE

student	
◦name	VARCHAR(40)
◦pob	VARCHAR(25)
◦dob	DATE
	age > 10 years old
◦sex	CHAR
	Only L or P
♦sid	CHAR(10)
	sid format is xx.yy.zzzz, x year of enrolment, y department code and z serial number

The CHECK constraints on a table are defined by associating CHECK constraints definition with the table when the table is created or altered. CHECK constraints are automatically activated when an INSERT or UPDATE statement modifies the data on the table. The CHECK constraints definitions used on the second testing are :

1) Constraint on *dob* will validate the age of students, the students age must have value greater than 10 years.
2) Constraint on *sex* column has limitation that only one character can be inputted that is *L* or *P* character. *L* represents male and *P* for female gender.
3) Constraint on *sid* will be limited that the student id inputted on this column must follow *xx.yy.zzzz* format. The *xx* represents the year enrollment in which each only contains the number between *0* up to *9* in two digits. The *yy* is department code, first *y* can only contain *0, 1, 2* or *5* number and second *y* is *1* or *2* number, so the possibilities of *yy* value are *01, 02, 11, 12, 21, 22, 51* or *52*. The *zzzz* represents serial number, each *z* can contain a number from *0* to *9*.

The SQL command to create Table II with CHECK constraints is displayed on Figure 3.

```
CREATE TABLE student (
name VARCHAR(40),
pob VARCHAR(25),
dob DATE,
sex CHAR,
sid CHAR(10) PRIMARY KEY
CONSTRAINT check_sid CHECK
(sid ~ '^[0-9]{1,2}\.[0125]?[12]?
  \.[0-9]{1,4}$')
CONSTRAINT check_sex CHECK
(sex IN ('L','P'))
CONSTRAINT check_age CHECK (
 DATE_PART('year', "dob"::timestamp) <
 DATE_PART('year', now() -
  interval '10 years'))
);
```

Figure 3. CREATE statement on a 4 constraints table

© 2014 ACADEMY PUBLISHER

Microsoft Access formatted data (*.accdb*) of student data from STMIK AMIKOM college are used for this research. The study is taken from only 1 table, namely the student table. The student table plays important role in AIS, the student table is used by AIS to record data from the first enrolment up to graduation of student. The study also takes only 5 columns of the student table from the data source according to the testing table. Comma separated value (CSV) is used to conduct testing on Table 1 and 2. Data source from Microsoft Access format can be export to CSV. CSV files are commonly used in importing files between applications. Microsoft Excel and Access, MySQL, Oracle and PostgreSQL are just few applications and databases capable for both importing and exporting CSV data [11]. This study only takes 1000 first records from 21439 Microsoft Access data records and is saved as CSV file named *data.csv*. An example of student data use in this study is shown on Figure 4.

```
'AKYUN SIGIT PRASETYO','BELITANG','25/08/1979','L','00.01.0746'
'AZMI ABDULKARIM HAYAZA','SEMARANG','03/11/1982','P','00.01.0791'
'MESAK MAURESY','TEPA','01/05/1981','L','00.01.0792'
'MUAMAR KADAAFI AZIZ','TEPA','30/07/1981','L','00.01.0793'
'ALBERT EINSTEIN ENUS','AMBON','24/10/1980','L','00.01.0794'
'RINES HEINS DAHOKLORY','TEPA','03/12/1982','L','00.01.0795'
'MOHAMAD ABDUH AL USAIRI','DURI','05/12/1980','L','00.01.0796'
'KETUT SUMARTAYASA','WANA BAKTI','30/04/1981','L','00.01.0797'
'YOANA FITRIANINGSIH','KUALA KAPUAS','31/07/1981','P','00.01.0798'
'ERNI WIDIASTUTI','TEMANGGUNG','10/05/1982','P','00.01.0799'
'SUSILO BUDI RAHARJO','NGAGLIK SLEMAN','21/12/1977','L','00.01.0800'
'IIN SRI RAHAYU','GILIMANUK','04/12/1981','P','00.01.0801'
'YANS SAFARID HUDHA','MAGELANG','29/12/1979','L','00.01.0802'
'TRI ENDARYANTI','KLATEN','30/08/1981','P','00.01.0803'
'PUSPITANING LUKITO','KLATEN','19/06/1982','P','00.01.0804'
'NOVIANA KARTIKA DEWI','TRENGGALEK','18/11/1981','P','00.01.0805'
'EKO BUDIYONO','WONOGIRI','09/01/1982','L','00.01.0806'
'ROSITA DEWI','WAKOLO','17/12/1983','P','00.01.0807'
'MUHAMMAD AROFIQ','TUBAN','26/11/1979','L','00.01.0808'
'NYOMAN ANDRIANI','SUMBAWA BESAR','11/04/1981','P','00.01.0809'
'MARINUS LAMBI','MANOKWARI','24/03/1982','L','00.01.0810'
'WAGIYO','KARANGKOBAR, BANJARNEGARA','18/02/1979','L','00.01.0811'
'AYUDHIA PERMATASARI','PATI','19/08/1982','P','00.01.0812'
'DIANA KURNIATY','JAKARTA','31/01/1982','P','00.01.0813'
'YANUAR HARTANTO','MAGELANG','22/01/1977','L','00.01.0814'
'ABDUL MUNIR','DOMPU','03/05/1979','L','00.01.0815'
'AMIR HUDA FAUJI','TULANGAN, SIDOARJO','13/06/1982','L','00.01.0816'
```

Figure 4. Student data on CSV format

On PostgreSQL, to input multiple records at once from external file using single statement (query) can be done by using 2 statements: INSERT or COPY. The format of INSERT statement to input 1000 records into student table for testing is shown on Figure 5.

```
INSERT INTO student VALUES
('student name1','place of birth1',
'date of birth1','sex1','sid1'),
('student name2','place of birth2',
'date of birth2','sex2','sid2'),
('student name3','place of birth3',
'date of birth3','sex3','sid3'),
.
.
.
('student name1000',' place of
 birth1000',' date of birth1000' ,
 'sex1000','sid1000');
```

Figure 5. INSERT statement for multiple rows

CSV file exported from Microsoft Access must be modified by adding a statement `INSERT INTO student VALUES` on the top of file. The bracket sign `(` must be added at every beginning of row, and sign `),` at every ending of row for each row except at the last row added with a `;` sign. The modified *data.csv* file is saved as *data.sql* file. The command `\i data.sql` is executed in a console of PostgreSQL to input the data. Before the data imported, PostgreSQL needs to adjust a date style format due date to style of data. Date of birth column on figure 4 shows data date style on DMY format. PostgreSQL supports the usual variety of standard SQL operator and functions defined by ANSI/ISO SQL standards, including date and time [12]. A DATESTYLE variable on PostgreSQL is used to set or show date and time. `SHOW DATESTYLE;` statement invoked from a console can be used to display the current value of DATESTYLE variable. The following statement sets the variable DATESTYLE to use DMY ANSI/ISO format, `SET DATESTYLE ="ISO, DMY";`. There are many errors while 1000 data being inputted into database especially on data formatting. There are many errors formatting on column *dob*, two type format dates are used on the data source. So when a table is defined to use specific format such as DMY, data having MDY format cannot be inputted. Most data used in this research use DMY format but some of them also use MDY format, as an example on Figure 6.

```
'LINDA YULIASTUTIK S.','BOYOLALI','2/11/03','P','00.01.0931'
'PIET ANTON PARDEDE','JAKARTA  19 April 1978','4/19/03','L','00.01.0958'
'DANIEL ALEXANDER','BALIKPAPAN','2/17/03','L','00.02.2877'
'DENI SATRIADI','LUBUK LINGGAU','2/18/03','L','00.02.2963'
'MURNIYATI','BATURAJA/12-05-1982','2/18/03','P','00.02.2995'
'DESI ANGGORO SETIYAWAN','DEMAK','2/18/03','L','00.02.3010'
'HAYUNINGTYAS HADI UTAMI','JAKARTA','2/18/03','P','00.02.3019'
'TINTUS WISNUWINARTO','PANUTAN','2/18/03','L','00.02.3023'
'YASMURI','TUBAN/27 Juli 1979','2/18/03','L','00.02.3025'
'SRI EKA LESTARI','JAKARTA','2/18/03','P','00.02.3079'
'ROI HENDRA BATUBARA','SIBOLGA','2/18/03','L','00.02.3086'
'RUDI RIVANDI','TANJUNG UBAN','2/10/03','L','01.01.1071'
'HERMAWAN CATUR PRASETYO','BANTUL','11/8/03','L','01.01.1079'
'JOKO PURWANTO','JAKARTA','6/18/03','L','01.01.1090'
'ANITA PUSPITA','TANJUNGPANDAN','6/30/03','P','01.01.1128'
'ANDI SARONO','SLEMAN','6/30/03','L','01.01.1150'
'SUGENG KUSBIANTORO','MANOKWARI','6/30/03','L','01.01.1156'
```

Figure 6. Different date format

The formatting error on date of birth is possible because of less validation at date input form by forcing input to use specific format that will cause incorrect data. Figure 6 shows that most of the date of birth on the data source are using MDY format that have 2003 (03) at the year of birth. If the student was born in the year of 2003, the data are definitely wrong because the database is data of student college, so the data possibility came from date of data being inputted. To prevent from this error can be done by limiting the minimal age of inputted student. Validation can calculate the inputted student age based on the inputted date of birth. Another error is that some data showing the inputted date of birth are inserted in the *pob* column, as displayed on Figure 6 on the 5^{th} and 9^{th} rows.

Besides INSERT statement, COPY statement can also be used to input multiple rows data at once in PostgreSQL. COPY statement uses CSV file or another user defines delimiters format as a data source. In this research, COPY statement can be directly used to import the data from *data.csv* above. COPY statement is run from PostgreSQL console as shown on Figure 7 to imported

© 2014 ACADEMY PUBLISHER

the data.

```
researchdb=> \copy student FROM
/home/emma/penelitian/data.csv
delimiters ',' CSV;
```

Figure 7. COPY statement

Both COPY and INSERT statements are run on 2 tables having different constraint description and are recorded in a time spent to perform the process. PosgreSQL has `\timing` command to calculated how long query spends. Executing `\timing` command on the console to enable or disable it. When that is enabled, each query run will include the amount of time taken at the end row.

After INSERT or COPY statement executed, data on the *student* table are deleted using TRUNCATE statement. INSERT and COPY statements are executed 20 times and taken 10 best time record. The testing flow process is shown on Figure 8.

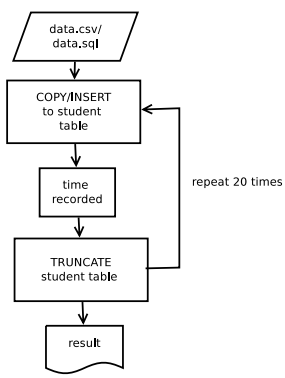

Figure 8. Testing Flow Process

This research also builds the interface program to input the data using web-based programming, HTML and PHP. Apache web server and PHP are used to connect PostgreSQL database server. The interface programs id used to test whether CHECK constraints give advantages although it is built without input data validation to restrict entered data. Figure 9 shows the interface program.

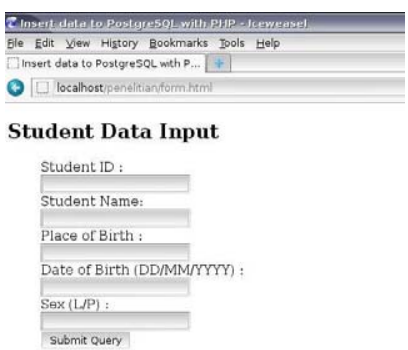

Figure 9. HTML FORM

In this test, data input cannot be done in multiple rows like previous research but it is only row by row. Figure 10 is source code from HTML FORM element which is used to build interface on Figure 9. Data inputted on that form are processed by PHP based program called *insert.php*.

```
<h2>Student Data Input</h2>
<ul>
<form name="insert" action="insert.php" method="POST" >
<li>Student ID :</li>
<li><input type="text" name="nim" /></li>
<li>Student Name:</li>
<li><input type="text" name="name" /></li>
<li>Place of Birth  :</li>
<li><input type="text" name="place" /></li>
<li>Date of Birth (DD/MM/YYYY) :</li>
<li><input type="text" name="date" /></li>
<li>Sex (L/P) :</li>
<li><input type="text" name="sex" /></li>
<li><input type="submit" /></li>
</form>
```

Figure 10. HTML FORM source code

PHP code program on *insert.php* file is to catch data from HTML FORM and then it is inserted into PostgreSQL shown on Figure 11.

```
<?php
$db = pg_connect("host=localhost port=5432
dbname=reserachdb user=emma password=123456");
$query = "INSERT INTO student VALUES
('$_POST[name]','$_POST[place]',
'$_POST[date]','$_POST[sex]','$_POST[sid]')";
$querya = "SET datestyle = 'ISO, DMY'";
pg_query($query0);
$result = pg_query($querya);
$result = pg_query($query);

if ($result === false) {
    print pg_last_error($db);
} else {
    print 'Succes';}
?>
```

Figure 11. PHP source code

IV. Discussion and Result

Migrating process from data source (*.accdb*) into *researchdb* on PostgreSQL faces many problems, such as different date format, incorrect age value on *dob* column or incorrect data on *pob* column. The migration process cannot directly import the data. The data source must be adjusted in order to be imported. The migration process shows how important of having qualified data is. Data validation can be a useful for maintaining data quality. Having qualified data will make data migration to the other system easier and CHECK constraint can help improving the data quality.

INSERT and COPY statements on the first experiment are done 20 times on each table and taken the 10 best time spent results. From the research result on inputting multiple rows using INSERT statement on single and multiple constraints table, there is a time difference between statements. Table III shows spent time difference between INSERT statement on single constraint table and multiple constraints table.

The Table III shows that INSERT command on the table without CHECK constraint reveals the best record

© 2014 ACADEMY PUBLISHER

TABLE III.
INSERT TESTING RESULT

	INSERT (ms)	INSERT with CHECK (ms)
	58.085	61.417
	48.515	57.826
	61.220	56.994
	64.651	52.708
	56.755	63.828
	58.939	60.001
	61.205	60.550
	56.672	59.462
	55.754	55.321
	57.605	63.069
Average	57.9401	59.1176

time is 48.515 ms and on the table with CHECK constraint is 52.708 ms. Graphical result of this testing is shown on Figure 12.

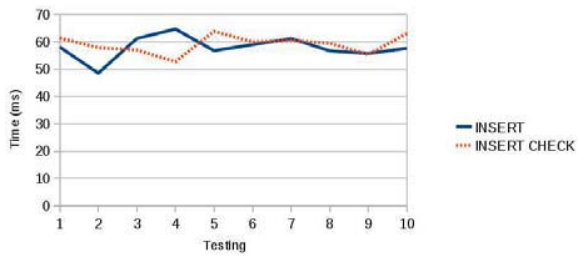

Figure 12. INSERT time comparison graphic

The spent time difference from COPY statement between single constraint and multiple constrains table is shown on Table IV and graphical result on Figure 13.

TABLE IV.
COPY TESTING RESULT

	COPY (ms)	COPY CHECK (ms)
	49.89	48.95
	49.835	55.518
	47.419	50.844
	46.471	54.098
	44.616	54.637
	41.742	55.675
	45.607	49.275
	48.913	46.966
	49.893	54.315
	43.398	50.404
Average	46.7784	52.0682

Table IV shows that COPY command takes the best time 41.742 ms on a table without CHECK constraint and 46.966 ms on a table with CHECK constraint. The time difference on the best time result on COPY command between table with and without CHECK constraint is 5.224 ms. This result also reveals that COPY statement is faster than INSERT statement for inputting multiple rows at single statement.

This study shows that there is time difference average: 1.2 ms ON input data using INSERT statement and 5.3 ms while using COPY statement between a table with

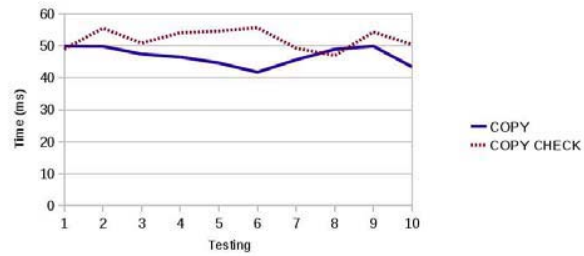

Figure 13. COPY time comparison graphic

single constraint PRIMARY KEY and a table added with 3 CHECK constraints. Considering the large of data (1000 rows) inputted at once, the results can be concluded that there is no significant difference between the input process by providing an additional CHECK constraint with no CHECK constraint on a table. In an AIS which contains a table of students as an example of the table used in this study, the input data of 1000 rows simultaneously is rarely done. Student data input process is generally done when the students start registering as a new student and the data will be not inputted in the large data at once.

This study also shows, that the provision of additional CHECK constraint on the student table is able to validate the entered data on the database level. CHECK constraint will reject data that do not comply with the table description. Even if the data directly inputted via the database console using INSERT or COPY statement, the data will be rejected. Figure 14 shows the rejection of a database system which prohibits the data input that does not comply with the table restrictions that are entered via database console.

```
researchdb=# INSERT INTO student VALUES
('Neisya Reehana','Sleman','12/5/2009','P','09.52.1205');
ERROR:  new row for relation "mahasiswa" violates check constraint "check_age"
researchdb=# INSERT INTO student VALUES
('Neisya Reehana','Sleman','12/5/2002','P','09.52.1205');
INSERT 0 1
researchdb=# INSERT INTO student VALUES
('Najwa Rashika','Sleman','12/08/2002','P','02.08.2002');
ERROR:  new row for relation "mahasiswa" violates check constraint "check_sid"
researchdb=# INSERT INTO student VALUES
('Najwa Rashika','Sleman','12/08/2002','P','02.52.2002');
INSERT 0 1
researchdb=# INSERT INTO student VALUES
('Naufal Rasendriya','Sleman','19/08/1999','X','08.52.1999');
ERROR:  new row for relation "mahasiswa" violates check constraint "check_sex"
researchdb=# INSERT INTO student VALUES
('Naufal Rasendriya','Sleman','19/08/1999','L','08.52.1999');
INSERT 0 1
```

Figure 14. The constraint that rejects INSERT

The first input data on Figure 14, *('Neisya Reehana','Sleman','12/5/2009','P','09.52.1205')*, shows that the age of an inputted student is less than 10 years old (born at 2009), so these data are rejected because they do not comply with the table description. If the age of student modified in order to comply with the table restriction, for example it is changed to the year of 2002 then the data will be accepted. Second example of data, (*'Najwa Rashika','Sleman','12/08/2002','P','02.08.2002'*), have incorrect student id format. The value of department code, 08 does not match with allowed code (01,02,11,12,21,22,51 or 52). If this value

© 2014 ACADEMY PUBLISHER

is changed to one of the allowed codes, such as 02.52.2002, thus the data can be inputted into the table. Third example on Figure 13, (*'Naufal Rasendriya','Sleman','19/08/1999','X','08.52.1999'*) is not accepted because the entered sex code (*X*) not eligible, the accepted sex code is only a character *L* or *P*.

The next experiment is conducting the simple interface program built from the HTML and PHP codes shown on Figure 10 and 11 to input the data on the table with CHECK constraints. The test results shown that even without validation on the interface program, the data cannot be accepted if they do not comply the table restrictions. An example of this test is shown on Figure 15, where the inputted data on *dob* column entered with today's date (25-4-2013) do not comply with the table restriction at *dob* column.

Data entered on Figure 15, rejected by PostgreSQL because the data do not comply with the age restrictions on the student table. The displayed error message shows that data violated a constraint (Figure 16).

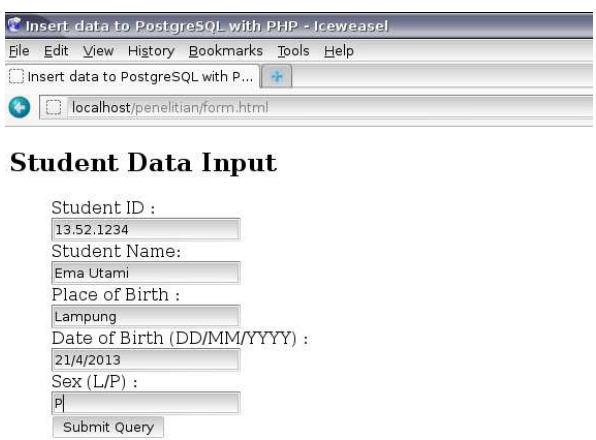

Figure 15. Input Data using HTML FORM

Figure 16 shows that the data will be rejected although the interface program does not use the input sanitization. The interface programmers do not need to make validation method, but they just can make FORM input and tailored the error message comprehended by the users. This advantage in turn eliminates the unintended programming errors. The data are kept valid and intact by the database itself.

When CHECK constraints are applied, any attempts to insert or update a row with an application or using a console database in *student* table with an invalid *sid, sex* or *dob* value, will cause the RDBMS to generate an error and the operations will fail. The main disadvantage of the CHECK constraint method is that it is hard to maintain. If frequent changes are needed to the information related to the allowable values on a column, then database administrators will have to change the SQL code in the CHECK constraint definition. In the AIS, *student* table column definition of *sid, dob* or *sex* are rarely changed, so adding CHECK constraints must be considered as a method to validate the data.

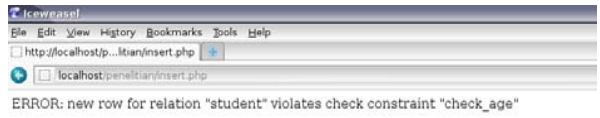

Figure 16. Error Message

V. Conclusions

Giving CHECK constraints on the database will be helpful to make a set of qualified data in which the data inputted can comply with the terms specified. Constraint at the database level greatly simplifies the interface programming of the database that does not need to think about how to make validations at the level of programming. Validation with the constraint database will maintain the validity of the database as the data which do not comply with the table restrictions could not be entered either although the data are inputted through the interface program or directly using SQL console. Having valid data will help to have a good quality data. The database with having strict table definition such as table of academic database use of CHECK constraint is very helpful to build qualified data.

The little time gap ranging from 1.2 ms to 5.3 ms for 1000 data entered at once on the table with only PRIMARY KEY constraint and table has 3 additional 3 CHECK constraints. The litle gap shows that introducing CHECK constraints on the database table is a necessity. Advantages gained by providing additional CHECK constraints for valid database and the easiness to make interface program are bigger than its disadvantages on the used speed of data input.

References

[1] R. Stair and G. Reynolds, *Principles of Information Systems*. Course Technology, Cengage Learning, 2011.

[2] V. S. Moertini, "Managing risks at the project initiation stage of large is development for hei: A case study in indonesia," *The Electronic Journal of Information Systems in Developing Countries*, vol. 51, 2012.

[3] S. Raharjo, "Constraint basis data sebagai fondasi yang kuat dalam pengembangan sistem informasi," 2012.

[4] L. Davidson and J. Moss, *Pro SQL Server 2012 Relational Database Design and Implementation*, ser. SpringerLink : Bücher. Apress, 2012.

[5] M. Schäler, T. Leich, M. Rosenmüller, and G. Saake, "Building information system variants with tailored database schemas using features," in *Advanced Information Systems Engineering*. Springer, 2012, pp. 597–612.

[6] T. Scholte, W. Robertson, D. Balzarotti, and E. Kirda, "Preventing input validation vulnerabilities in web applications through automated type analysis," in *Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual*. IEEE, 2012, pp. 233–243.

[7] C. Coronel, S. Morris, and P. Rob, *Database Systems: Design, Implementation, and Management [With Access Code]*. Course Technology/Cengage Learning, 2012.

© 2014 ACADEMY PUBLISHER

[8] P. G. D. Group, *Postgresql 9 0 Official Documentation - Volume I the Sql Language*, ser. Linux Documentation Library. Fultus Corporation, 2011, no. v. 1.
[9] R. Press, "Improving data quality in relational databases: Overcoming functional entanglements," 2011.
[10] P. Nielsen, *SQL Server 2005 Bible*, ser. Bible. Wiley, 2006.
[11] W. Gilmore, *Beginning PHP and MySQL: From Novice to Professional*, ser. Apresspod Series. Apress, 2010.
[12] J. Drake and J. Worsley, *Practical PostgreSQL*. O'Reilly Media, 2011.

Ema Utami received the Bachelor, Master and Doctoral degrees in Computer Science from Gadjah Mada University, Yogyakarta, Indonesia in 1997, 2002 and 2010 respectively. Since 1998 she has been a lecturer in STMIK AMIKOM Yogyakarta, Indonesia. Her areas of interest are Natural Language Processing, Computer Algoritms, and Database Programming.

© 2014 ACADEMY PUBLISHER