"instances" seems like the wrong word here (for example, "combination of existing instances" doesn't really make sense grammatically).
I like "kinds of variation", although it will obviously require some other rewording.
I think these sentences can also be simplified to something like (feel free to use this as a starting point):

Variational databases are intended to support any kind of variation in databases. Many specific kinds of variation in databases have already been well-studied and are well-supported, for example, by work on schema evolution and data integration. However, many other kinds of variation, and combinations of different kinds of variation, are not well addressed by existing work.

# AN ABSTRACT OF THE DISSERTATION OF

Parisa Ataei for the degree of Doctor of Philosophy in Computer Science presented on June ?, 2021.

Title: Theory and Implementation of a Variational Database Management System

Abstract approved: _____

Eric Walkingshaw

In this thesis I present the variational database management system, a formal framework and its implementation for representing variation in relational databases and managing variational information needs. The variational database is intended to support any instance of variation in database, both the existing instances and more importantly newly arisen instances which could be a combination of existing instances. For example, existing instances of variation in databases are when a single database evolves over time and when data from different sources is combined. While there are context-specific solutions to both problems neither solution can address when data sources combined in one database evolves over time.

a query language for variational databases

This thesis collects a large amount of work: It defines the variational database framework and the syntax and [specific kind of] semantics of the variational relational algebra. It also defines the variational information needs that must be satisfied in a generic variational database framework and [shows/proves] that the introduced framework satisfies all these needs. It illustrates two usecases of the variational database framework that represent two existing instances of variation in databases. It presents the variational database management system which is the implementation of the variational database and variational relational algebra ~~and is an abstract layer~~ on top of a traditional RDBMS. It also presents several theoretical results related to the framework and query language, such as syntax-based equivalence rules that preserve the semantics of a query, a type system for ensuring that a variational query is well formed with ~~regards~~ to the underlying variational schema, and a confluence property of variational relational algebra type

"variational information needs" is the wrong term -- I would say VRA lets us express variational information needs

what about just "properties"?

use cases

as an abstraction layer written in Haskell

respect

system and semantics with ~~regards~~ respect to the relational algebra type system and semantics.

# Theory and Implementation of a Variational Database Management System

by

Parisa Ataei

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented June ?, 2021
Commencement June 2021

Doctor of Philosophy dissertation of Parisa Ataei presented on June ?, 2021.

APPROVED:

---

Major Professor, representing Computer Science

---

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

---

Parisa Ataei, Author

# ACKNOWLEDGEMENTS

[Eric. Committee. jeff. abu. parents. friends. ]

# TABLE OF CONTENTS

# TABLE OF CONTENTS (Continued)

# LIST OF ALGORITHMS

Algorithm Page

# Chapter 1: Introduction

[points: - def variation in db and is everywhere - instances but context specific solution don't suffice - thus a generic framework that addresses problems]

[ - dimension, behaviour - instances but context specific solution don't suffice - example of instances - well-studied: schema evolution - partial: SPL - new instance either out of combination of other or new - generic framework to instantiate for each instance and address all variational needs.]

## 1.1 Motivation and Impact

[motivation]

### 1.1.1 Motivating Example

[combination of instances, behaviours, and dimensions]

## 1.2 Contributions and Outline of this Thesis

[contribution]

# Chapter 2: Background

[background]

## 2.1 The Relational Database

[relational database]

## 2.2 Relational Algebra

[relational algebra]

## 2.3 Variational Set

[vset]

### 2.3.1 Variational Set Configuration

[vset configuration.]

## 2.4 The Formula Choice Calculus

[formula choice calculus]

# Chapter 3: The Variational Database Framework

[needs. must have configuration. ]

## 3.1 Variational Needs in a Relational Database

[needs and examples of them.]

## 3.2 Variation Space in a Variational Database Framework

[fexp. evaluation.]

## 3.3 Variational Schema

[vsch]

### 3.3.1 Variational Schema Configuration

[vsch configuration.]

## 3.4 Variational Table

[vtab]

### 3.4.1 Variational Table Configuration

[vtab configuration]

## 3.5 Variational Database

[vdb]

### 3.5.1 Variational Database Configuration

[vdb configuration]

## 3.6 Properties of a Variational Database Framework

[well-formed vdb properties.context-specific properties.]
[show that they hold for vdb.]

# Chapter 4: The Variational Query Language

[vql]

## 4.1 Variational Relational Algebra

[vra]

### 4.1.1 VRA Configuration

[vra configuration]

### 4.1.2 VRA Semantics

[vra semantics]

### 4.1.3 VRA Type System

[type sys]

### 4.1.4 VRA Variation-Minimization Rules

[rules]

## 4.2 Variational Query Language Properties

[prop. show for vra.]

# Chapter 5: Variational Database Management System (VDBMS)

[vdbms]

## 5.1 Implemented Approaches

[apps]

## 5.2 Experiments

[exp.]

# Chapter 6: Related Work

[related work! have to work on this!]

## 6.1 Instances of Variation in Databases

[schema evolution. database versioning. data integration. data provenance.]

## 6.2 Instances of Database Variation Resulted from Software Development

[SPL. data model. query.]

## 6.3 Variational Research

[blah]

# Chapter 7: Conclusion

[conclusion]

# APPENDICES

# Appendix A: Variational Database Usecases

## A.1 Variation in Space

[enron email usecase]

## A.2 Variation in Time

[employee evolution usecase]