
CONTENTS

APPROVAL	ii
SELF DECLARATION AGAINST PLAGIARISM	iii
ABSTRACT	iv
ABSTRAK	v
DEDICATION	vi
ACKNOWLEDGMENTS	vii
PREFACE	viii
CONTENTS	ix
LIST OF TABLES	xi
LIST OF FIGURES	xii
1 INTRODUCTION	1
1.1 Background of the Study	1
1.2 Research Question	3
1.3 Objectives	3
1.4 Hypotheses	4
1.5 Scope And Limitations	4
2 REVIEW OF LITERATURE AND STUDIES	5
2.1 Object-Relational Mapping (ORM)	5
2.1.1 Comparison of Popular Python ORM Frameworks	5
2.1.2 Just-In-Time (JIT) Compilation	6
2.2 Related Studies	6
2.3 Research Gap	7
3 RESEARCH METHODOLOGY	9
3.1 Research Design	9
3.2 JIT Development	12
3.2.1 Justification for Choosing llvmlite	14
3.2.2 Compilation Trade-offs	15
3.3 Testing And Evaluation	15
3.4 Data Collection Procedure	16

3.5	Data Analysis Techniques	17
4	PERFORMANCE ANALYSIS	19
4.1	Testing Environment and Dataset Overview	19
4.2	Performance Comparison	21
4.2.1	Execution Time Analysis (512MB, 0.2 CPU Core)	22
4.2.2	CPU Usage Analysis (512MB, 0.2 CPU Core)	23
4.2.3	Memory Usage Analysis (512MB, 0.2 CPU Core)	24
4.2.4	Execution Time Analysis (1024MB, 0.5 CPU Core)	25
4.2.5	CPU Usage Analysis (1024MB, 0.5 CPU Core)	26
4.2.6	Memory Usage Analysis (1024MB, 0.5 CPU Core)	27
4.2.7	Single-Record Operation Analysis (1024MB, 0.5 CPU Core)	27
4.2.8	Real-World Scenario: Retrieving Comments	29
4.3	Unexpected Results and Revalidation Attempts	32
4.4	Summary of Performance Insights	32
4.4.1	Performance Impact of JIT Integration (Answer to RQ1)	32
4.4.2	JIT Efficiency by Access Pattern (Answer to RQ2)	33
4.4.3	Framework Integration and Evaluation(Answer to RQ3)	34
4.4.4	Integration of JITORM	36
4.4.5	Integration of SQLAlchemy	38
4.4.6	Integration of Pony ORM	40
4.4.7	Integration of Tortoise ORM	42
4.4.8	Implementation Considerations for Adopting JITORM	43
5	CONCLUSION	45
5.1	Conclusion	45
5.2	Recommendations	46
5.3	Limitations	46
5.4	Final Remarks	47
	Appendices	51
A	Curriculum Vitae	52